Practical Electronics

The UK's premier electronics, computing and maker magazine

Max's Cool Beans
Digital Etch A Sketch &
the classic Snake game

Teach-In 12
Part 6: RF testing
and measurement

Audio Out
All-pass filters:
unsung heroes

Circuit Surgery
Stabilising op amps
using output resistors

BAT+
BAT–
TEST BATT 8.6V
BATTERY: 0.0V
MAX CURR: 1A
POWER
DOWN
UP
ENTER
MODE
Versatile Battery Checker
- LiPo 602535
+ 3.7V 400mAh
SILICON CHIP ©2023
PICO ANALYSER RevB
diymore
Pico 2

Power Rail Probe shows supply noise and ripple

Pico 2 Audio Analyser

Test batteries, large and small!

PLUS!

Techno Talk: optical MEMS microphones and nanoscale sensors

Our new **Power Electronics** series will cover DC/DC, AC/DC and DC/AC conversion, power factor correction (PFC) and more

Apr 2026 £5.99

www.electronpublishing.com | @practicalelec | practicalelectronics

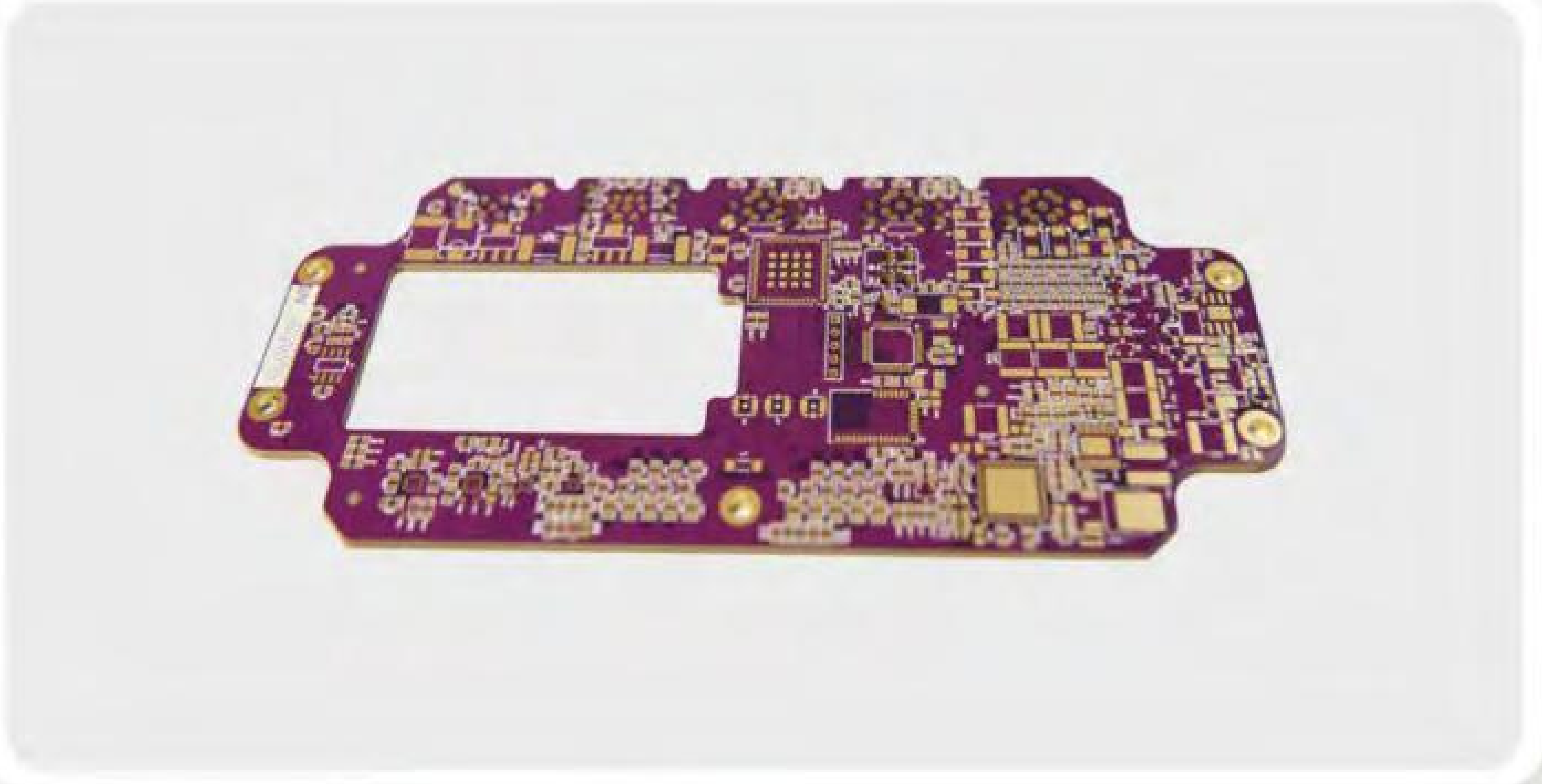
PCBWay

Beyond Standard —

Discover PCBWay's Advanced PCB Lineup

About PCBWay:

With dedicated workshops for advanced PCBs, PCBWay offers full-spec, high-precision, and large-scale production. We ensure customers are supplied with PCBs that meet stringent quality standards at the lowest cost.

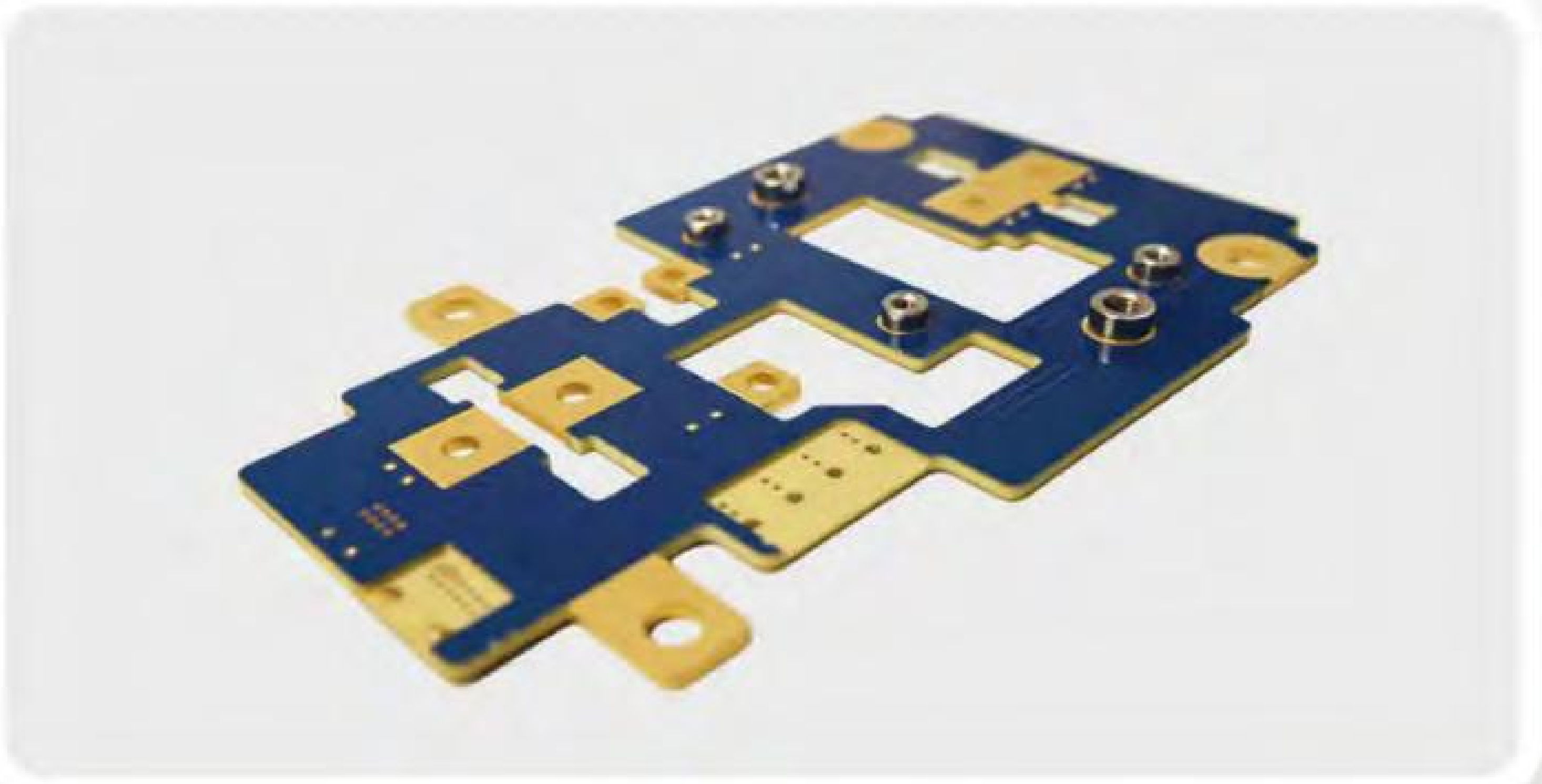


HDI PCB //

- 4-60 layers
- FR-4, High Tg, Halogen-free, PTFE
- Finished Copper: 1/3OZ–8OZ
- Min Track/Spacing (inner): 2/2mil (H/HOZ base copper)
- Min BGA Pitch: 0.35mm

High Frequency PCB //

- RO4003C, RO4350B, Ro3003, Ro3010, RT5880
- Min. Track/Spacing: 2mil
- Min. Hole Size: 0.15mm
- Finished Cooper: 1-2oz
- Board Thickness: 0.2–3.2mm



Other PCBs with Special Process //

- Embedded Copper PCB
- Backplane PCB
- Cavity PCB
- White/Partially Hybrid PCB
- Black Core PCB (After-dark)

www.pcbway.com service@pcbway.com

Practical Electronics

Volume 55. No. 04
April 2026
ISSN 2632 573X

Contents

INPUT
50 kΩ
Maximum
±50 V
OUTPUT
50 Ω
LOAD
Power Rail
Probe
POWER
CHARGE
OFF - ON

Projects and Circuits

Versatile Battery Checker by Tim Blythman 14
This useful tool allows you to check the condition of all manner of batteries, including Li-ion, LiPo, lead-acid, SLA, AGM, alkaline and zinc-carbon types. It can test 12V or 24V vehicle starter batteries, 9V batteries, plus D, C, AA and AAA cells and more.

Power Rail Probe by Andrew Levido 33
This is one of those pieces of test equipment that you don't really need, until you do. In conjunction with an oscilloscope, it allows the accurate measurement and evaluation of ripple, switching noise and transients riding on DC supply rails.

The Pico 2 Audio Analyser by Tim Blythman 60
Using an inexpensive Raspberry Pi Pico 2 microcontroller board and not much else, we can generate and analyse audio signals. It has oscilloscope and spectrum modes and can run a sweep to plot a frequency response or perform harmonic analysis.

Series, Features and Columns

Teach-In 12.6: the World of Wireless by Mike Tooley 4
This month, we will be looking at the principles and techniques that underpin a range of useful RF measurements, including voltage, power and frequency. Our Hands-On project is a handy RF probe.

Power Electronics, part one by Andrew Levido 24
Power electronics broadly describes circuits designed to handle electrical energy. We will explore power electronics in this new series, including some practical examples.

Max's Cool Beans by Max The Magnificent 40
Weird & Wonderful Arduino Projects, part 16: etching sketches & snake shenanigans
After looking at one reader's console built from MDF, we will first create an electronic Etch A Sketch, then start implementing the classic arcade game Snake.

Circuit Surgery by Ian Bell 48
Measuring circuit frequency response using a PC sound card, part 7: op amp output impedance & compensation. This instalment considers the effect of adding a series output resistor on stability, especially when driving capacitive loads.

Techno Talk by Max the Magnificent 58
From visible mechanisms and fixable machines to sealed laptops, microscopic sensors and optical MEMS. Join me on a ramble through modern technology as it grows more powerful and more mindboggling.

Audio Out by Jake Rothman 69
Absolutely phased out (all-pass filters, part one): most people know about low-pass, high-pass, band-pass and notch filters; so what are all-pass filters and are they useful?

Regulars and Services

Editorial 2
The benefits of desktop computers

Reader services – Editorial and Advertising Departments 2

Subscribe to Practical Electronics and save money 3

Practical Electronics Back Issue DOWNLOADS – 2024 now available! 13

Practical Electronics PCB Service 76-78
PCBs for Practical Electronics projects

Classified ads and Advertiser index 79

Next month! – highlights of our next issue of *Practical Electronics* 80

Made in Australia & the UK.

Written in Britain, Australia, the USA and Ireland.

Read everywhere.

© Silicon Chip Publications Pty Ltd 2026

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The May 2026 issue of *Practical Electronics* will be published on Thursday, 2nd of April, 2026 (see p80).

Renewals paid by PayPal, Visa and Mastercard are now handled via our website:
www.electronpublishing.com

Practical Electronics

Volume 55. No. 04
April 2026
ISSN 2632 573X

Editorial

Editorial offices
Electron Publishing (Australia)
Tel +61 2 9939 3295
Email pe@pemag.au
Web www.electronpublishing.com

Address mail to:
Electron Publishing (Australia)
PO Box 194, Matraville NSW 2036
Australia

Advertising enquiries
+61 2 9939 3295 pe@pemag.au

Editor Nicholas Vinen
Publisher Nicholas Vinen
Digital subscriptions Stewart Kearn **Tel 07918 614662**
Online Editor Alan Winstanley
Web Systems Kris Thain
Production Bao Smith
Technical staff Tim Blythman, John Clarke

Print subscriptions
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom
Tel 01202 087631
Email pesubs@selectps.com

Technical enquiries
We regret that technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine.

Questions about articles or projects should be sent to the editor by email: pe@pemag.au

Projects and circuits
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

Some projects and circuits published in *Practical Electronics* employ voltages that can be lethal. Do not build, test, modify or fix any mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies
Silicon Chip Publications may offer kits or other parts for making our projects, but not in all cases. When kits are not available, readers will need to find and source parts themselves. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements
Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

Transmitters/bugs/telephone equipment
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

The benefits of desktop computers

These days, when someone says they use a "PC" (which I would say includes Macs and Linux machines), they usually mean a portable computer – what we once called a laptop, and now more often a notebook.

Portability obviously has its advantages. Still, whenever I work on a notebook computer, I feel as though I'm operating with one hand tied behind my back compared to a proper desktop system.

The most obvious limitation is screen space. My eyesight is still serviceable, but not what it once was, and trying to work efficiently on a 43cm diagonal (17-inch) display inevitably involves squinting at small text and constantly rearranging windows. By contrast, large 4K desktop monitors (eg, 110cm/43-inch) are now inexpensive and transform productivity; even more so if you have space for two screens.

Yes, you can connect a notebook to an external monitor – possibly via a dock – but once you do that, it's no longer particularly portable. And the display is only part of the problem.

Keyboards are another major compromise. A full-size keyboard with decent key travel is far more comfortable for extended typing than even the best notebook keyboards. Personally, I rely on a split ergonomic keyboard, which lets me work all day without sore hands or wrists; something no notebook keyboard has ever managed.

Pointing devices are similar. Modern trackpads may be adequate for casual use, but they're no substitute for a proper mouse. Again, you can plug one in, but that adds clutter and requires desk space, further eroding the supposed advantages of portability.

I do miss the TrackPoint used on Lenovo (and earlier IBM) notebooks. It remains the best non-mouse pointing device I've ever used, and took up virtually no space.

Finally, there's thermal performance. Notebook computers are inherently thermally limited. Even when fitted with powerful CPUs, they tend to throttle quickly under sustained workloads; the sort of workloads involved in software development, simulation, PCB layout, CAD, desktop publishing, graphic design or data processing.

For my own work, including editing, PCB layout and coding, I simply get more done in the same amount of time on a desktop computer. I often wonder whether people who rely exclusively on notebooks have found ways around these productivity limitations, or whether they just don't realise what they're missing.

Nicholas Vinen, Electron Publishing (Australia)*
* a division of Silicon Chip Publications Pty Ltd.

Practical Electronics

UK readers SAVE £1 on every issue

SUBSCRIBE NOW!

01202 087631

Variable-Speed Drive for Induction Motors
Digital Capacitance Meter
High-Bandwidth Differential Probe
An IR Keyfob Transmitter Programmed via NFC
Power Rail Probe shows supply noise and ripple
Versatile Battery Checker
Pico 2 Audio Analyser
Test batteries, large and small!

Take out a one-year subscription and save more than £10 over the year.
Even better – save over £1 per issue if you subscribe for two years – a total saving of £26.

Overseas rates represent exceptional value

You also:
- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Guarantee a copy, even if your local newsagent sells out**

Order by:
- **Phone or post with a cheque, postal order or credit card**
- **Or order online at: www.electronpublishing.com**

Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £33.99, 12-months £59.99, 24-months £114.99

Europe Airmail: 6-months £38.99, 12-months £69.99, 24-months £129.99

Rest Of The World Airmail: 6-months £46.99, 12-months £84.99, 24-months £164.99

Cheques (in **£ sterling only**) payable to ***Select Publisher Services Ltd*** and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

Also via our online shop at**: www.electronpublishing.com**

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £29.99 for 12 months, more details at: **www.electronpublishing.com**

SUBSCRIPTION ORDER FORM

- ☐ **6 Months: UK £33.99, Europe £38.99 (Airmail), Rest of the World £46.99 (Airmail)**
- ☐ **1 Year: UK £59.99, Europe £69.99 (Airmail), Rest of the World £84.99 (Airmail)**
- ☐ **2 Years: UK £114.99, Europe £129.99 (Airmail), Rest of the World £164.99 (Airmail)**

To: **Practical Electronics Subscriptions**
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

I enclose payment of £ ....
(cheque/PO in £ sterling only)
payable to ***Select Publisher Services Ltd***

☐ Please charge my Visa/Mastercard

My card number is: ....
Please print clearly, and check that you have the number correct

Card Security Code .... Valid From Date.......................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....

Name ....

Address ....

Post code .... Tel. ....

Email ....

Subscriptions start with the next available issue.

Teach-In 2026

by Mike Tooley

World of Wireless – An Introduction to Radio and Wireless Technology

Series 12, part 6: RF testing and measurement

In the last instalment, our focus shifted to antennas, including how several popular types are built, how they work and how well they perform. We investigated several basic antennas and explained the importance of matching them properly. Our Hands-On project featured a 60-300MHz dipole reference antenna.

This month, we will be looking at the principles and techniques that underpin a range of useful RF measurements, including voltage, power and frequency. Our Hands-On project is a handy RF probe for use with an ordinary DC meter for RF measurements over a range extending from 100kHz to 100MHz.

Measuring RF voltages

If a signal is within its measurement range, the easiest method of measuring RF signals is with the aid of an oscilloscope and compensated probe. Reliable measurements of peak-to-peak voltage can usually be made at frequencies up to and beyond 30MHz. Even the budget hand-held oscilloscope shown in Fig.6.1 can produce accurate indications up to 10MHz.

If you are lucky enough to possess a high-specification bench 'scope, you may be able to make measurements up to 100MHz or higher.

Where frequencies are beyond the measurement range of a 'scope, a simple diode detector (like that shown in Fig.6.2) can be used in conjunction with a conventional analog or digital meter display. The detector responds to the peak value of the RF input, but the meter is often scaled in root-mean-square (RMS) units (see later).

If you look carefully at Fig.6.2 you should be able to correlate each of the surface-mounted components with those shown in the circuit diagram, Fig.6.3 (there are two diodes in one package).

The low-cost diode detector shown in Fig.6.2 is specified for operation from 100kHz to 3.2GHz in a 50Ω system via the SMA input connector.

To minimise any stray reactance (that would otherwise degrade a detector's high-frequency performance), the RF voltmeter's probe tip and ground connection need to be kept very short. Commercial wideband RF voltmeters overcome this problem by using probe mounts to extend the frequency range well into the UHF/SHF range.

At signal levels below 1V, it's important to be aware that the DC output voltage from a simple diode detector responds to the peak value of the applied RF voltage less the diode's forward voltage drop. This results in nonlinearity at low input levels, as illustrated in Fig.6.4.

Where a digital readout is required, a low-cost microcontroller with integral analog-to-digital converter (ADC) can be employed along the lines shown in Fig.6.5. Based on an Arduino Uno and a 16×2 I2C LCD screen, this delightfully simple arrangement only needs a connection to one of the Uno's analog input ports from the diode detector shown in Fig.6.6.

The code fragment shown in Listing 6.1 (later in the article) shows how to cope with the RMS scaling and conversion to dbV (more on that later).

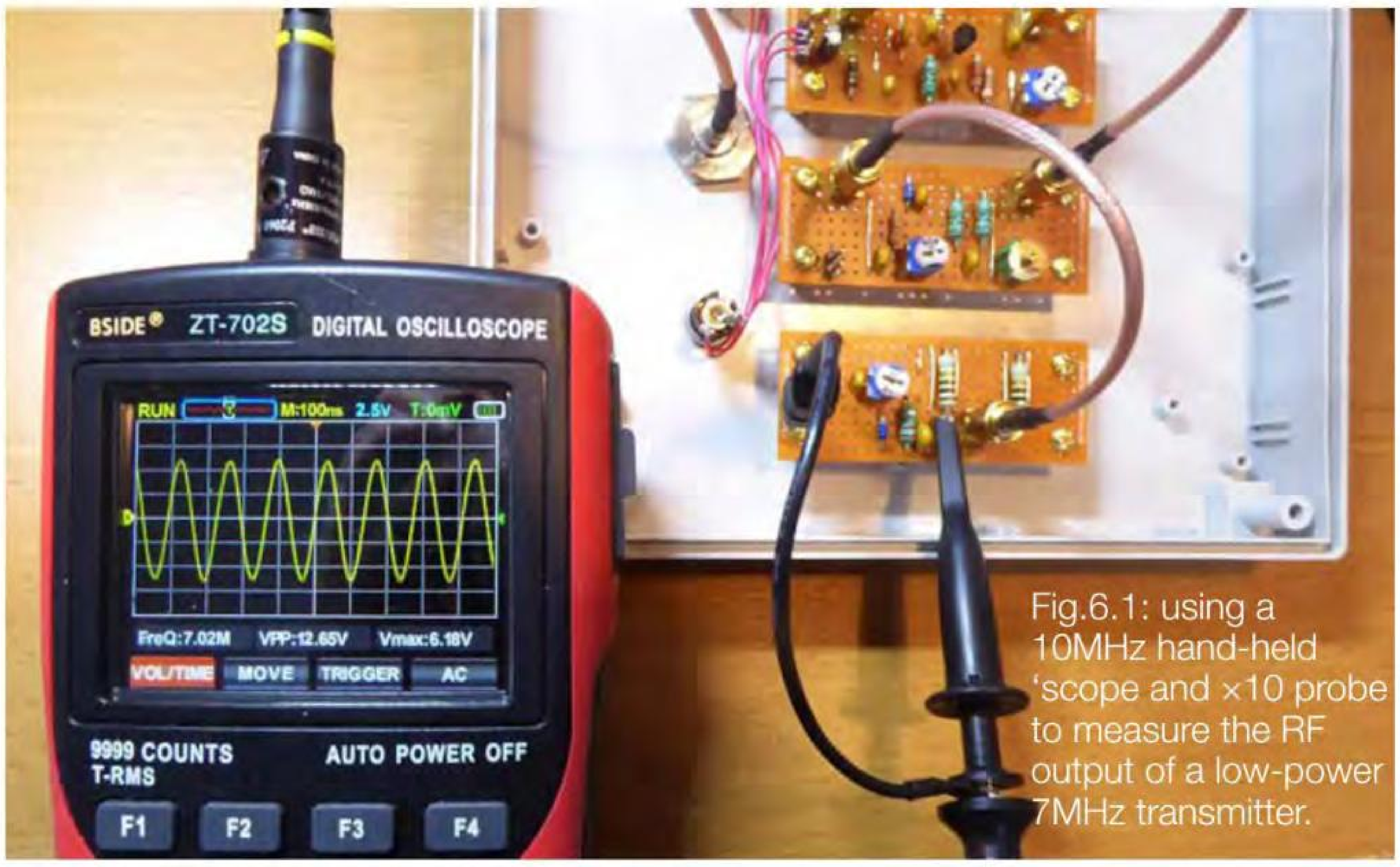


Fig.6.1: using a 10MHz hand-held 'scope and ×10 probe to measure the RF output of a low-power 7MHz transmitter.

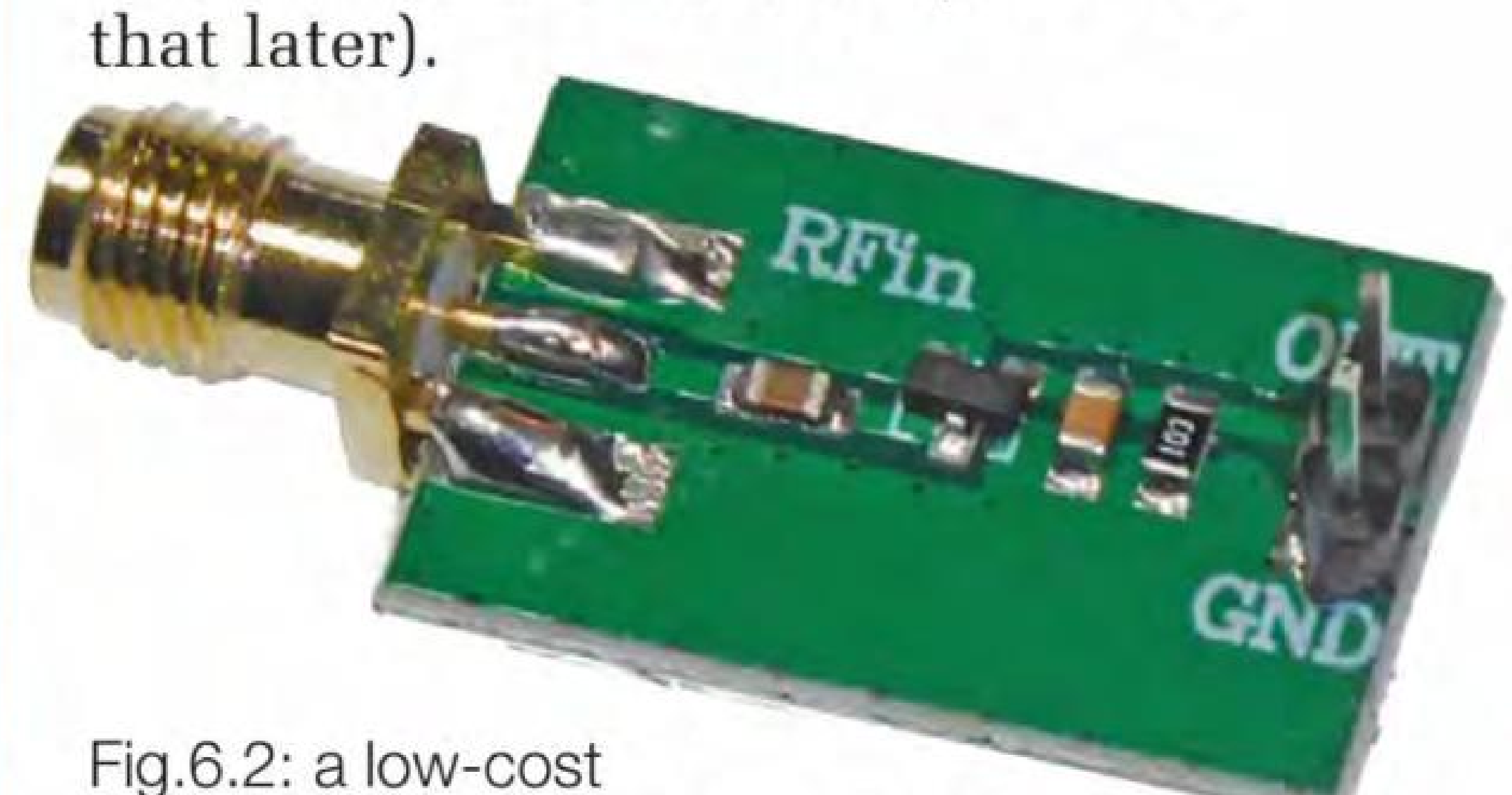


Fig.6.2: a low-cost diode-based RF detector module.

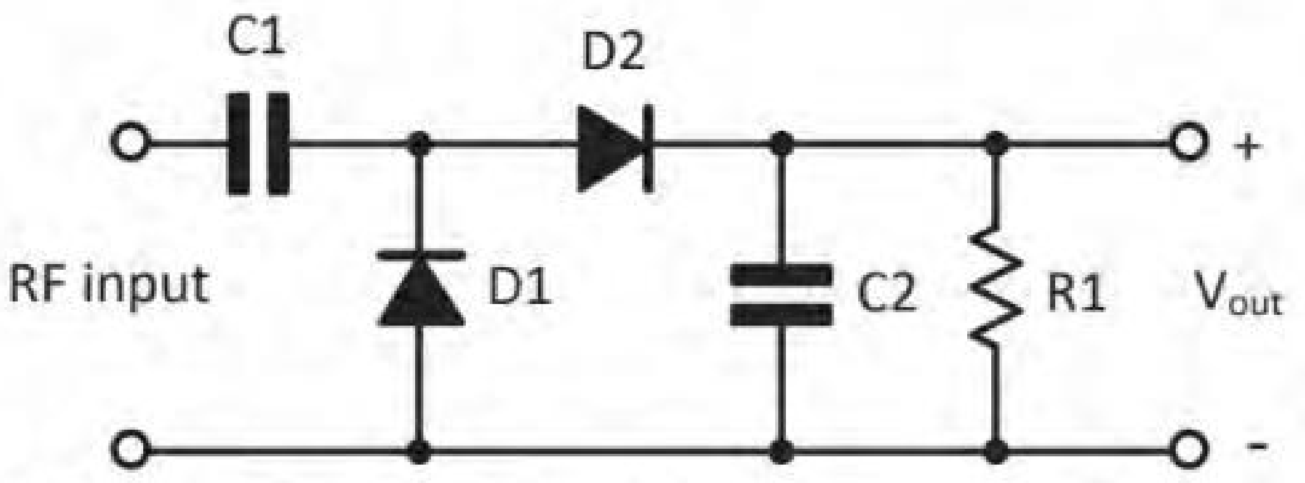

C1
D2
RF input
D1
C2
R1
+
V_{out}
-

Fig.6.3: the circuit of the low-cost diode-based detector module.

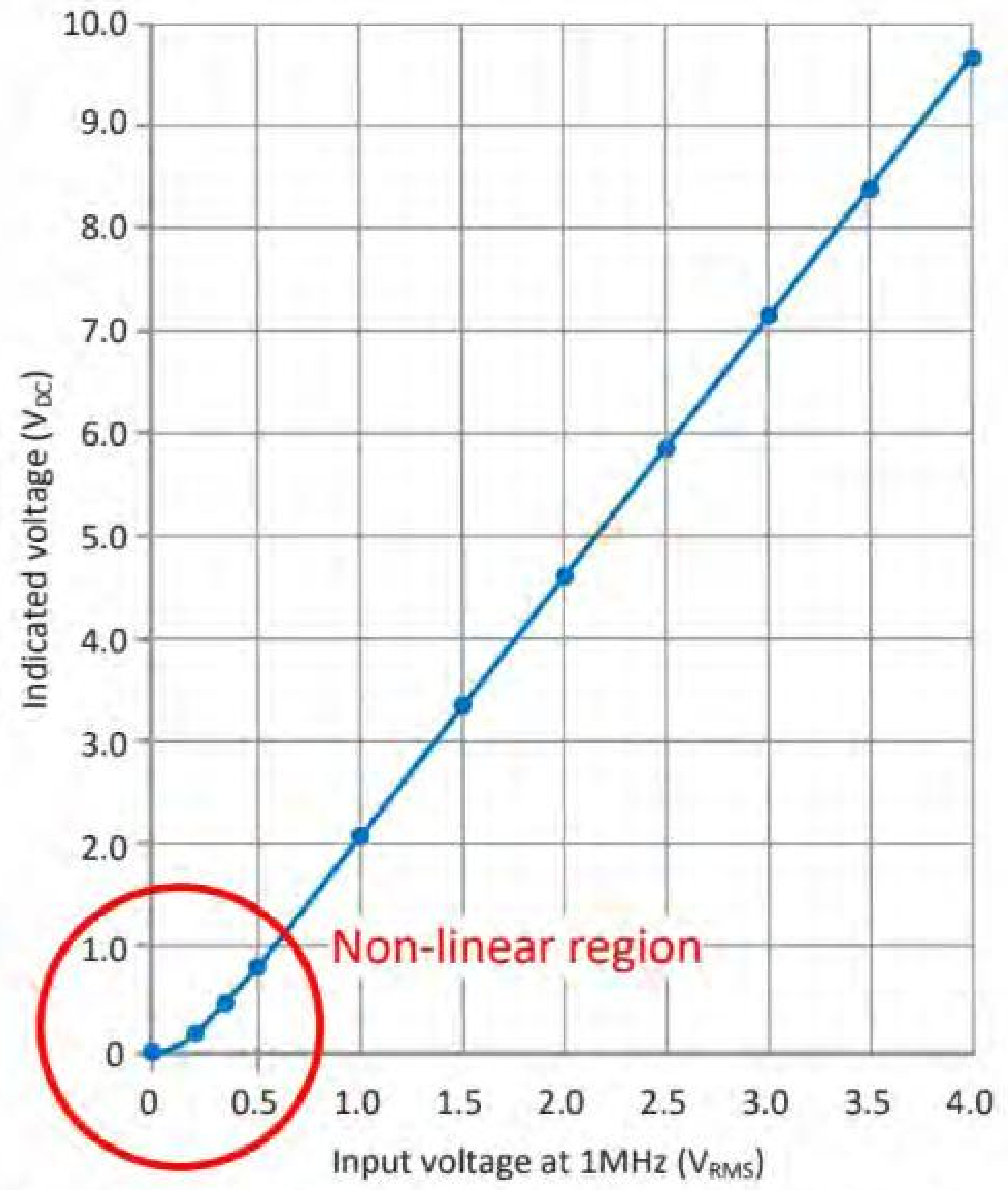


Fig.6.4: a plot of output voltage vs RMS input voltage for the Fig.6.3 detector.

Limitations

When interfacing with a microcontroller, it's important to avoid applying too much voltage to the analog input ports. Fig.6.6 avoids this problem with R1 and D3 protecting the input when more than 5V is applied to an analog input port. Note also that the RF input must be limited so that the maximum reverse voltage of the detector diodes is not exceeded.

In Fig.6.6, each of the BAT42 diodes in the voltage doubler arrangement has a maximum reverse voltage rating of 30V, so the maximum input voltage that can be safely applied to the voltage doubler arrangement is about 20V RMS. For greater input voltages, the BAT42 diodes should be replaced with BAT41 diodes (100V maximum reverse voltage).

Alternatively, a compensated attenuating probe can be used.

Voltmeter calibration & sensitivity

Simple diode detectors measure the peak RF voltage and display RMS values assuming the signal is a pure sinewave, making their readings inaccurate for other waveform shapes. True RMS voltmeters instead calculate the effective AC value by determining the square root of the average squared instantaneous values over a full cycle, regardless of waveform shape.

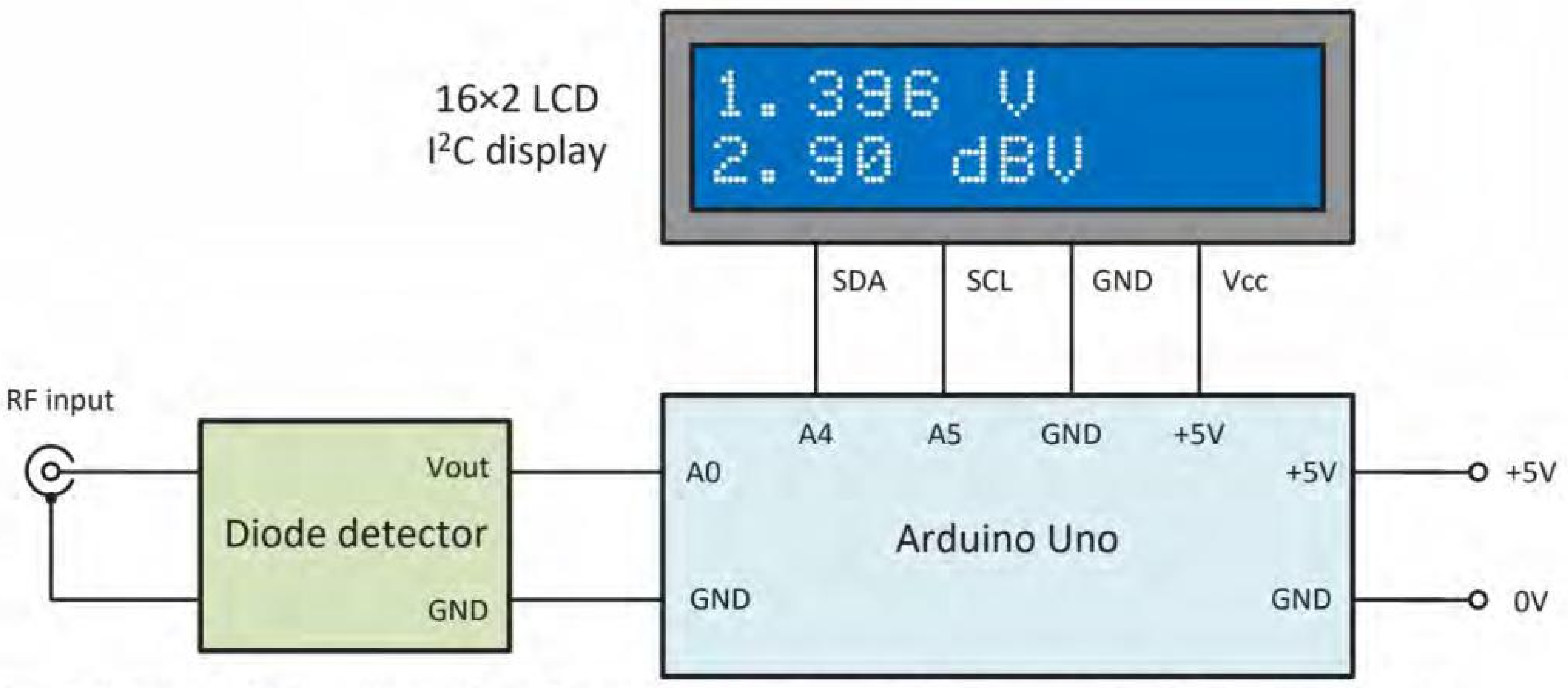

16×2 LCD
I²C display
1.396 V
2.90 dBV
SDA
SCL
GND
Vcc
RF input
Vout
Diode detector
GND
A4
A5
GND
+5V
A0
+5V
Arduino Uno
GND
GND
+5V
0V

Fig.6.5 an RMS-reading RF voltmeter based on an Arduino and 16×2 LCD screen.

Devices like the popular Analog Devices AD8318 offer accurate true RMS detection through dedicated circuitry.

The sensitivity limitations inherent in a simple diode detector can be addressed by employing a specialised integrated circuit such as the AD8318. It is designed for precise relative received signal strength indication (RSSI) and automatic power regulation.

The AD8318 utilises multiple amplifier stages and detectors to compress and sum the input signal, delivering a DC output that reliably represents the RMS value across an extensive frequency range, from 50MHz to over 3.5GHz. The device features nine cascaded amplifiers, each with an individual detector (see Fig.6.7), with outputs that are collectively summed to generate the final DC output voltage.

The AD8318 has a dynamic range of 65dB but, because of the high voltage gain, care must be taken to ensure that the input power is kept below 16mW (corresponding to an input voltage of 0.89V_{RMS} in a 50Ω system). An interior view of the author's home-built AD8318 RF detector head is shown in Fig.6.8.

Using decibels for voltage

Decibels provide us with a means of specifying quantities using a logarithmic rather than linear scale. This helps keep numerical values within a reasonable range. Expressed in decibels, the ratio of two voltages, V_1 and V_2, is calculated as:

$$n = 20\log_{10}\left(\frac{V_2}{V_1}\right)\text{dB}$$

When used to specify a relative voltage (rather than gain or loss), we use a reference value, V_{ref}, as the denominator, ie:

$$n = 20\log_{10}\left(\frac{V}{V_{ref}}\right)\text{dB}_{\text{rel}}$$

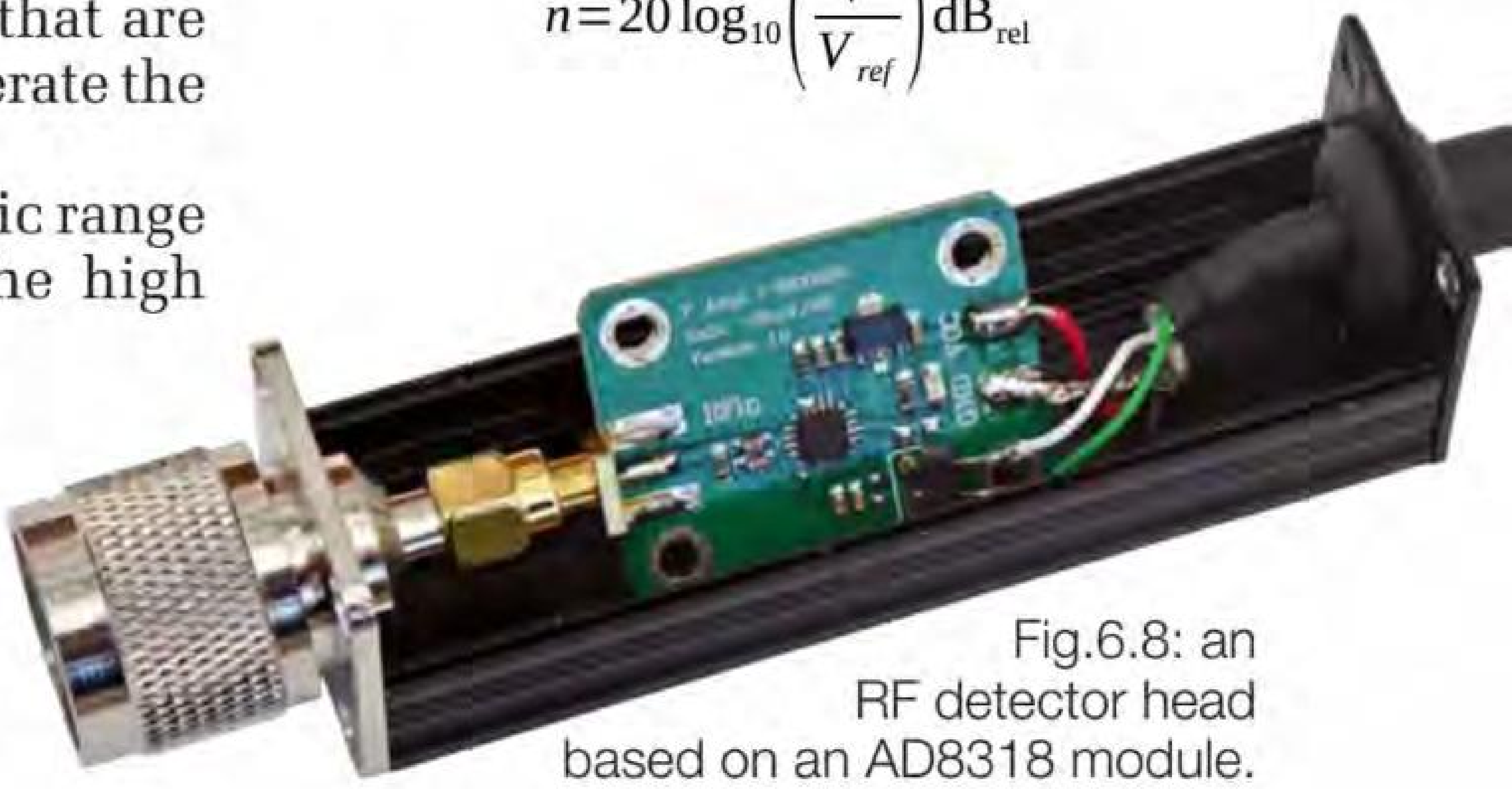


Fig.6.8: an RF detector head based on an AD8318 module.

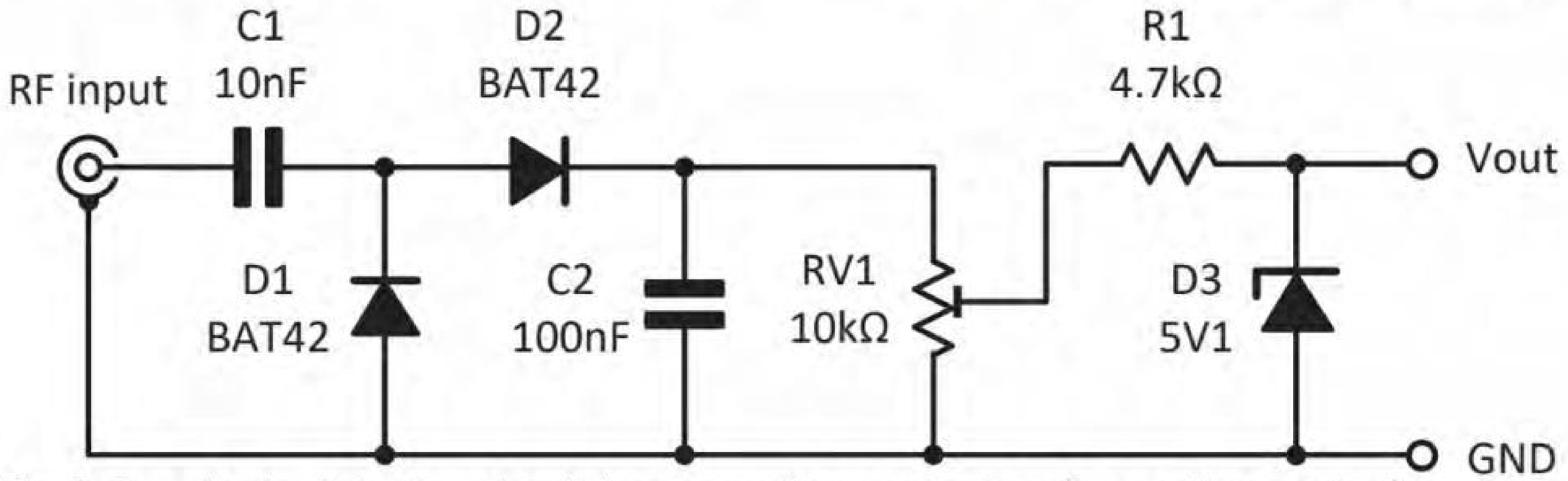

RF input
C1
10nF
D2
BAT42
R1
4.7kΩ
Vout
D1
BAT42
C2
100nF
RV1
10kΩ
D3
5V1
GND

Fig.6.6: a diode detector circuit incorporating protection for a microcontroller.

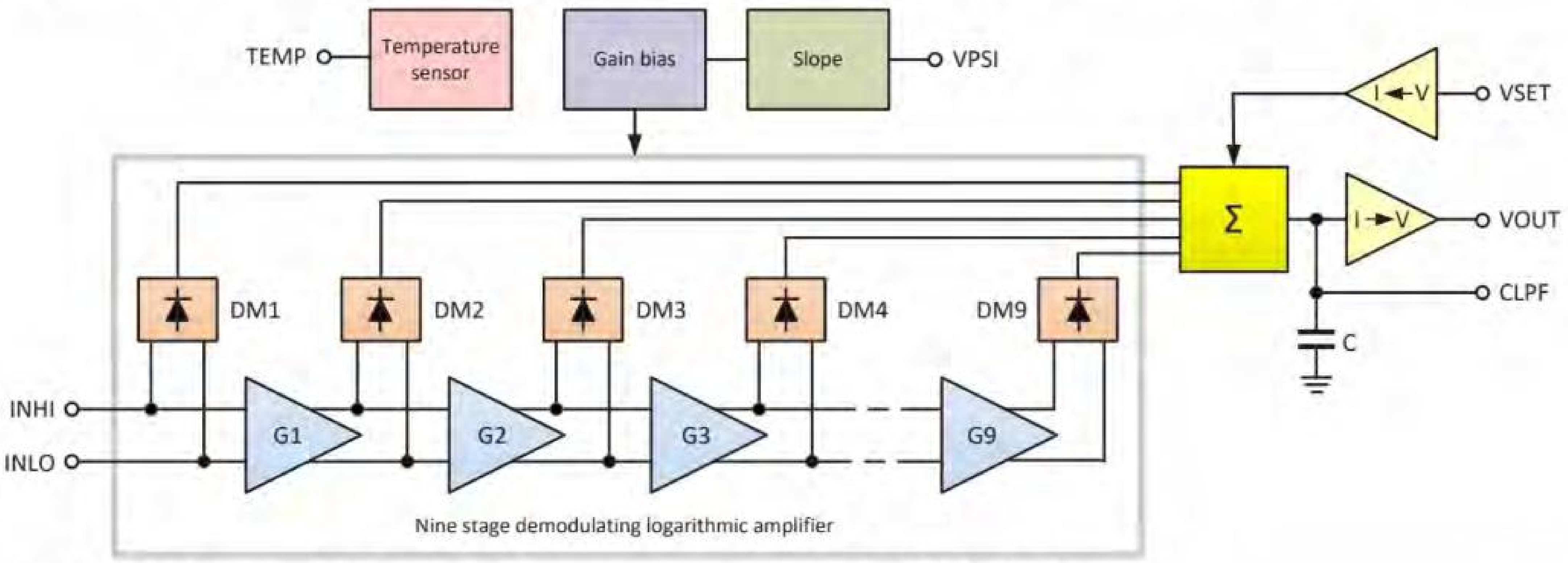

TEMP
Temperature sensor
Gain bias
Slope
VPSI
VSET
I→V
Σ
VOUT
CLPF
C
DM1
DM2
DM3
DM4
DM9
INHI
INLO
G1
G2
G3
G9
Nine stage demodulating logarithmic amplifier

Fig.6.7: the simplified internal arrangement of an AD8318 IC.

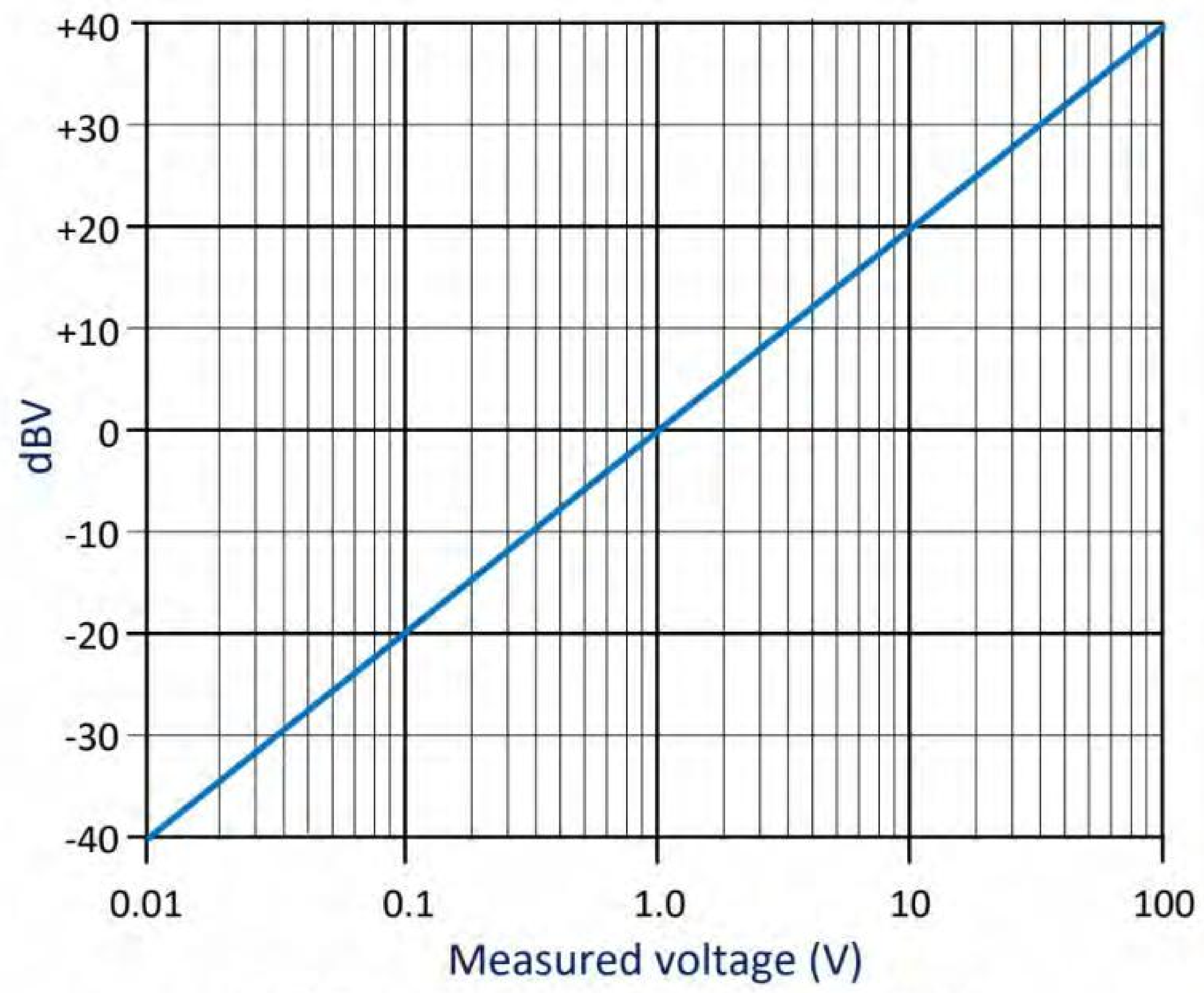


Fig.6.9: a plot of dBV against RMS voltage.

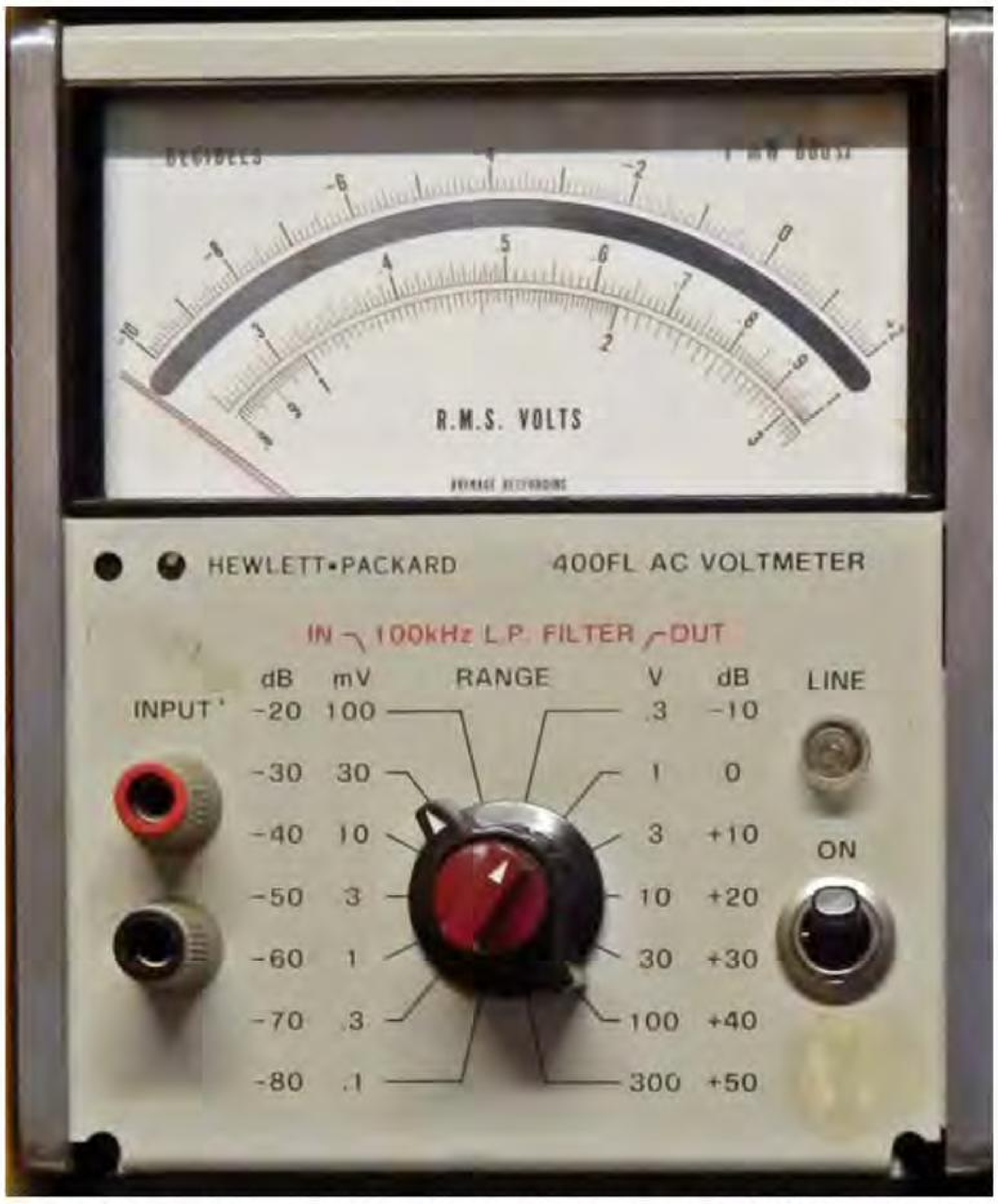


Fig.6.10: a wideband AC voltmeter calibrated in RMS volts & dBV.

1V is often used as the reference value, in which case the relationship becomes:

$$n = 20\log_{10}\left(\frac{V}{1\,\text{V}}\right)\text{dBV} = 20\log_{10}(V)\,\text{dBV}$$

To help you understand this relationship, Fig.6.9 shows dBV plotted against RMS voltage. Note that 1V = 0dBV, 10V = 20dBV, 100V = 40dBV and so on. Fig.6.10 shows a typical meter display with separate scales for RMS voltage and dBV.

The reason that the formulae above use a scaling factor of 20, rather than 10 (as you might expect from the unit name decibel), is that decibels are fundamentally a measure of power ratio, not voltage ratio. When comparing voltages across the same impedance, power is proportional to the square of voltage: $P = V^2 \div R$.

Taking the logarithm of a squared voltage gives $2\log_{10}(V)$, and when this is multiplied by 10 (the 'deci' part of decibel), the result is the familiar $20\log_{10}(V)$ scaling.

Measuring RF power

Several different technologies are used for measuring RF power. They include:

- thermal methods, in which the heating effect of RF power in a matched load is sensed using a thermistor (or 'bolometer')
- using diode detectors to rectify the RF voltage across a matched load, producing a DC output voltage proportional to signal amplitude
- applying the attenuated RF signal from a matched attenuator to an integrated circuit that senses the true RMS value of input voltage from the attenuator
- sampling the RF signal as a baseband AC signal, which can then be directly digitised, processed digitally and displayed
- calculations of power based on voltage measurements made with a wideband oscilloscope, compensated probe and matched resistive load

Both thermal and detector-type measurements are often described as 'direct sensing' because the conversion of an RF signal to a DC output is performed by means of a probe or head mounted close to the signal source. More complex 'indirect' methods involve a multi-stage conversion performed remotely, where amplitude data is scaled to provide a power reading.

Thermal power sensors usually employ thermistors attached to a matched load. The resistance of the thermistor sensor changes with load temperature. This change is detected using a temperature-compensated auto-balancing bridge. The output of the bridge is a DC voltage that can be displayed using an analog meter or converted using an ADC (eg, within a microcontroller) for digital display.

Due to the intermediate conversion of RF energy to heat, thermal sensors provide a true RMS response over an exceptional range of frequencies. Unfortunately, while they are excellent for measuring average power, they can be too slow for some applications (eg, they will barely respond to brief high-level bursts of RF energy due to thermal inertia).

Simple diode detectors provide a fast response, but they can be insensitive without additional amplification. Due to the non-linearity of their characteristic and maximum reverse voltage rating, diode detectors have a limited dynamic range.

The modern solution to the measurement of RF power is with the aid of integrated circuits designed specifically for true RMS voltage and power measurement. Supported by a matched attenuating load and software running on a microcontroller or PC, these devices provide a sensitive and highly cost-effective solution to RF power measurement at frequencies up to several gigahertz.

Using an oscilloscope

At lower RF frequencies, it is possible to use an oscilloscope to measure RF voltage and thus, provided

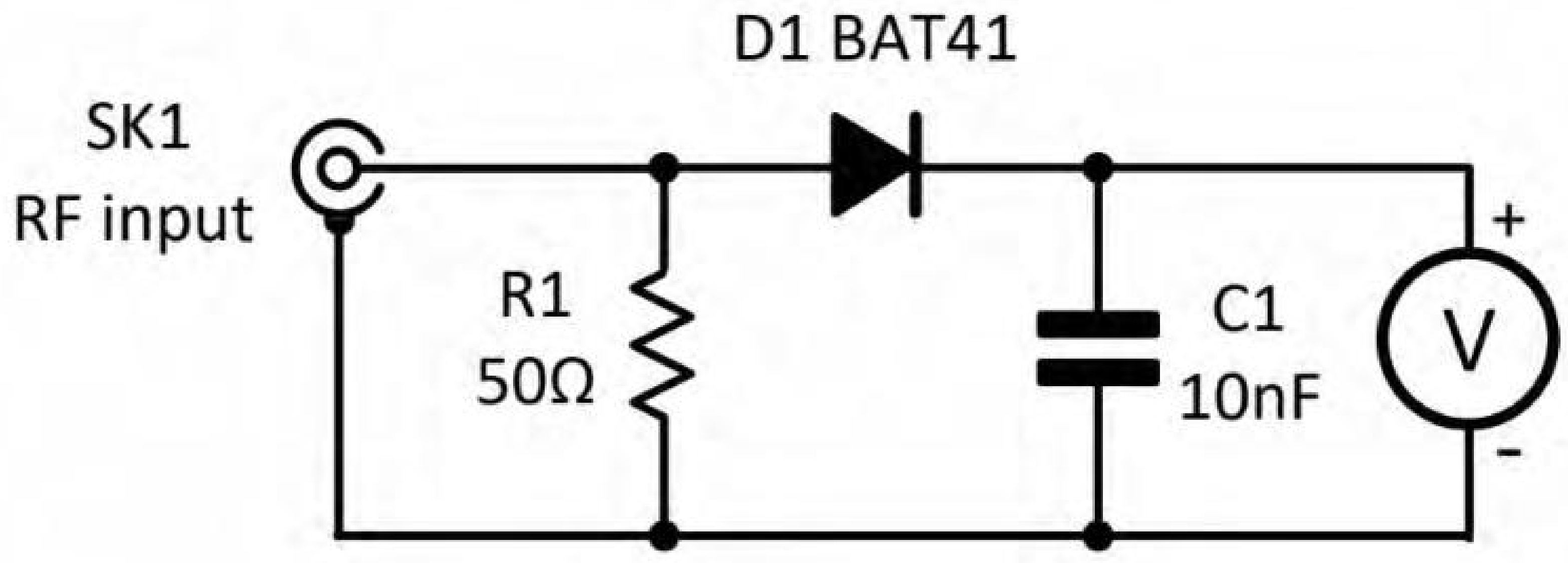


Fig.6.11: the circuit of a basic RF power meter.

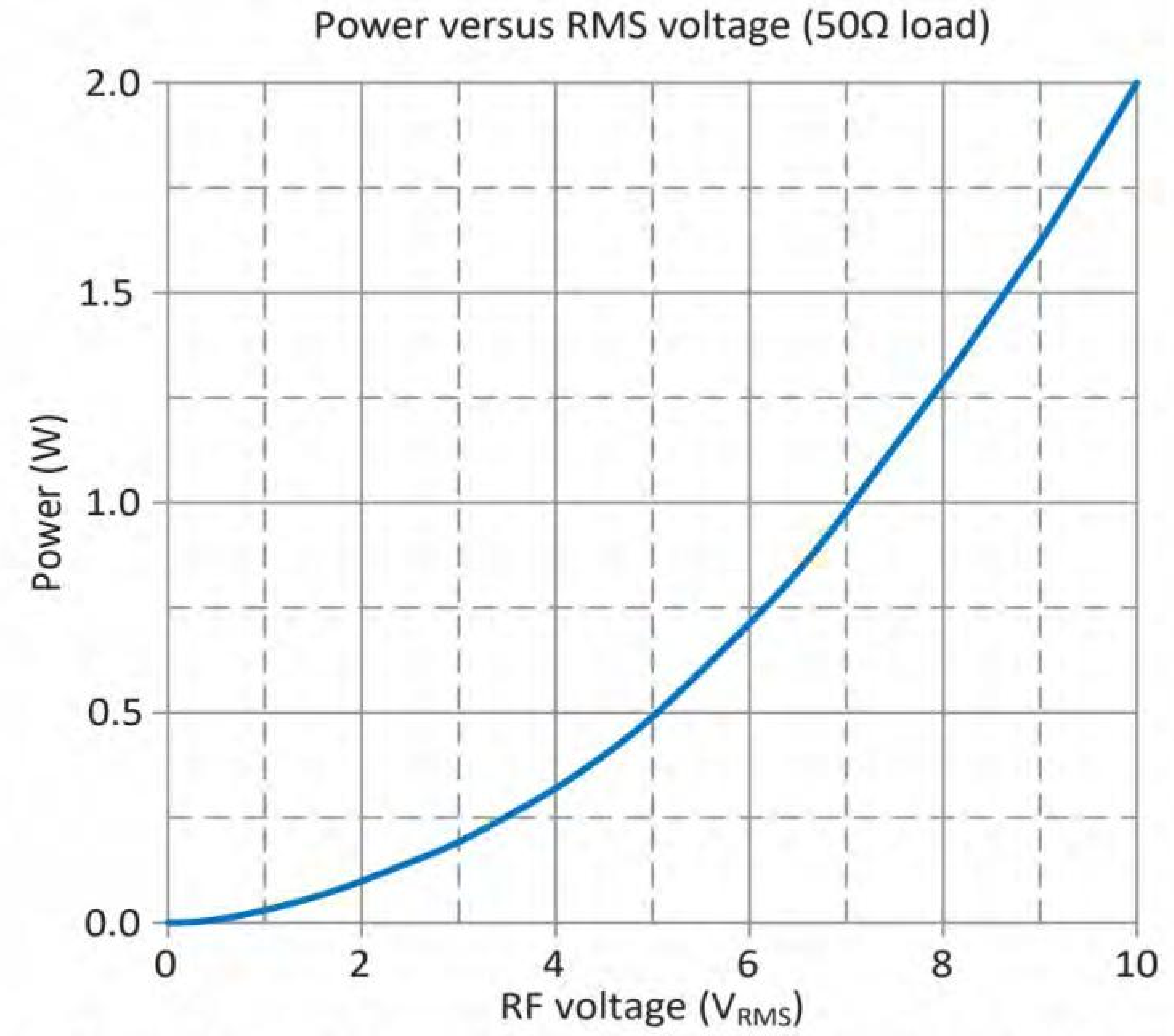


Fig.6.12: the relationship between power and voltage for Fig.6.11.

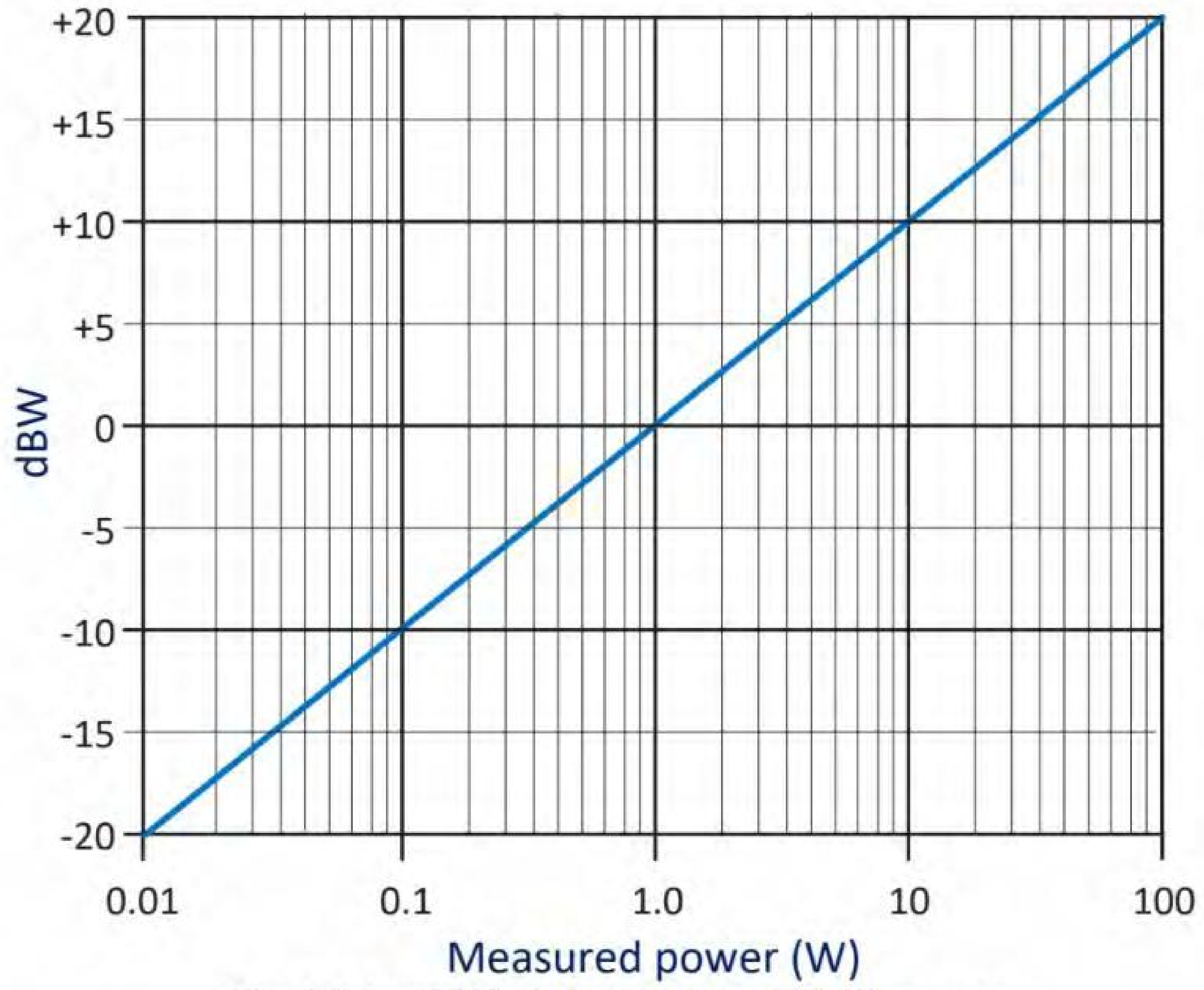


Fig.6.14: dBW plotted against RMS power.

that the load resistance is accurately known, to determine power from the peak-to-peak value of a waveform observed on an oscilloscope (many oscilloscopes can also perform accurate RMS measurements). The following relationship is used:

$$P_{out}=\frac{\left(V_{out(pk-pk)}\right)^2}{8R_L}\ \text{(in watts)}$$

Putting this into context, the 'scope display shown earlier in Fig.6.1 indicates a peak-to-peak voltage of 12.65V. This voltage is being developed across a 50Ω load consisting of two 100Ω 1W resistors connected in parallel. The output power can thus be calculated as follows:

$$P_{out}=\frac{V^2_{out(pk-pk)}}{8R_L}=\frac{12.65V^2}{8\times 50\Omega}=\frac{160V^2}{400\Omega}=0.4\,\text{W}$$

If the waveform is not a sinewave (or has a lot of distortion) and your oscilloscope has a built in RMS voltage measurement feature, you can use that to get an accurate RMS measurement and then convert the voltage to a power figure using the load resistance. Some high-end oscilloscopes may have a mathematical function to perform that conversion for you.

In that case, ensure you have a sufficient number of signal cycles visible on the screen for an accurate RMS measurement. If you can see, say, one-and-a-half cycles of the signal, the calculation may be inaccurate due to the imbalance that comes from including only half of the second cycle.

Standalone power meters

For regular measurements, a dedicated power meter is a better alternative to an oscilloscope, with the added advantage that many standalone RF power meters incorporate a good-quality 50Ω load. Such instruments may have digital or analog displays and are usually calibrated in watts (W) or milliwatts (mW).

The circuit of a basic RF power meter is shown in Fig.6.11. This simple arrangement incorporates a suitably rated 50Ω resistive load across which the applied RF voltage is measured. Fig.6.12 shows the square law relationship between power and applied voltage in Fig.6.11. The resulting nonlinear calibration of a typical RF power meter scale is shown in Fig.6.12.

Fig.6.13 shows a typical example of power measurement at RF using a combined power and SWR meter and an external load. The meter's digital display indicates an output power of exactly 2W.

Using decibels for power

Decibels are frequently used to specify power in RF systems. As with voltage, this helps keep numerical values within a reasonable range. Expressed in decibels, the ratio of two powers, P_1 and P_2, is calculated from:

$$n=10\log_{10}\left(\frac{P_2}{P_1}\right)\text{dB}$$

When used to specify a relative power (rather than gain or loss), we use a reference value, P_{ref}, as the denominator, as follows:

$$n=10\log_{10}\left(\frac{P}{P_{ref}}\right)\text{dB}_{\text{rel}}$$

1W is often used as the reference value, in which case the relationship becomes:

$$n=10\log_{10}\left(\frac{P}{1\,\text{W}}\right)\text{dBW}=10\log_{10}(P)\,\text{dBW}$$

This relationship is illustrated graphically in Fig.6.14, where 1W = 0dBW, 10W = 10dBW, 100W = 20dBW and so on. Note that the

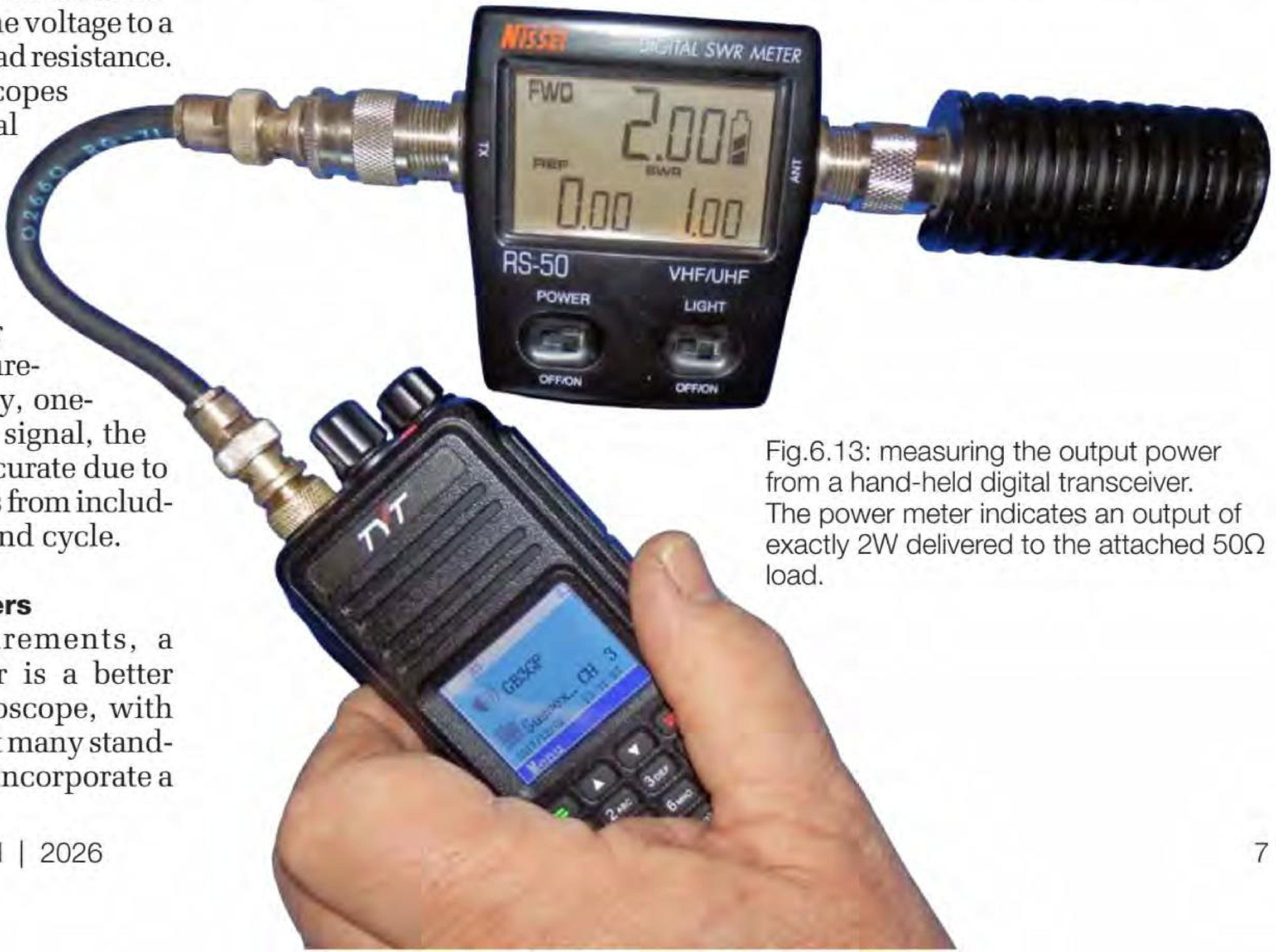


Fig.6.13: measuring the output power from a hand-held digital transceiver. The power meter indicates an output of exactly 2W delivered to the attached 50Ω load.

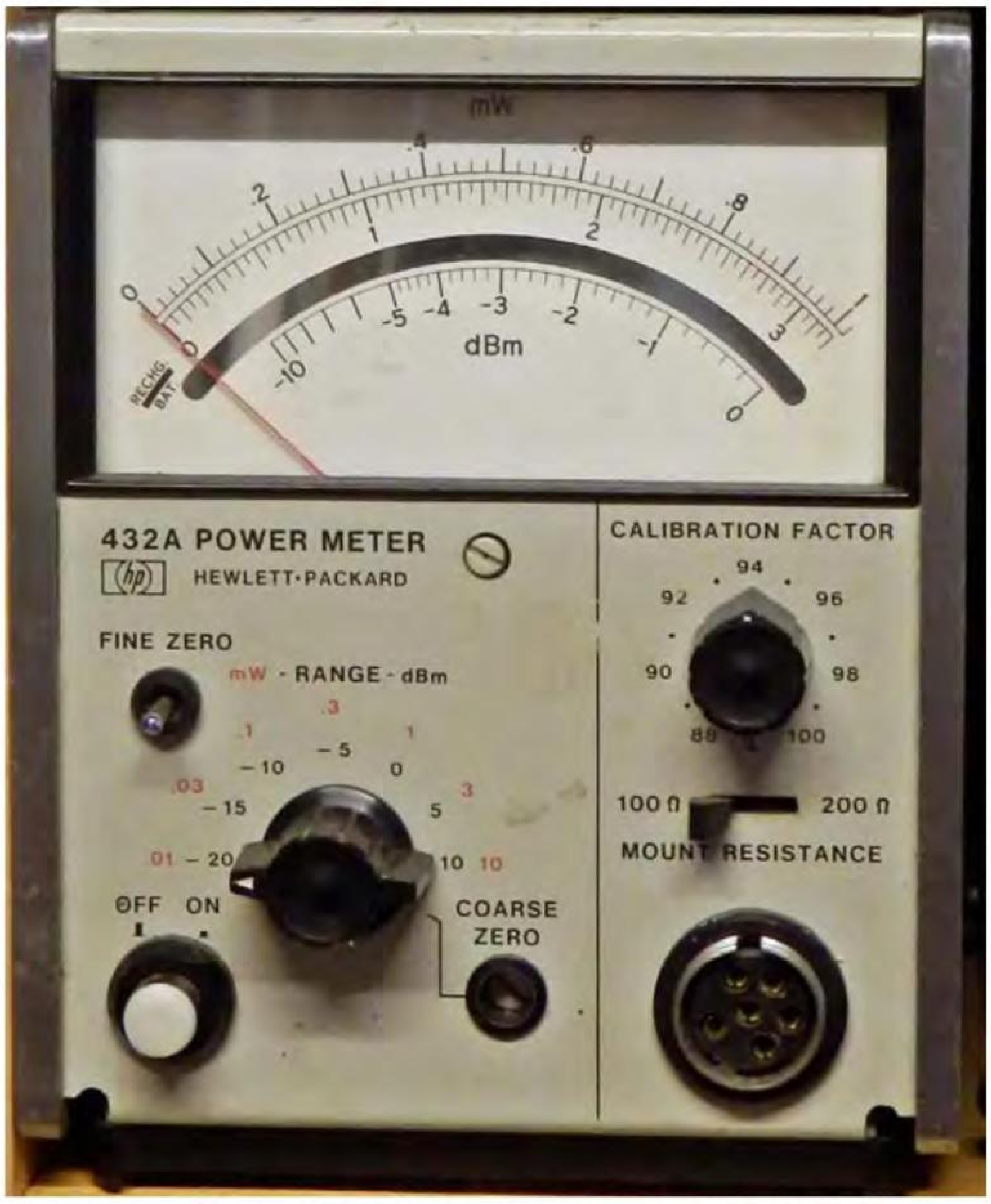


Fig.6.15: an RF power meter calibrated in RMS power and dBm.

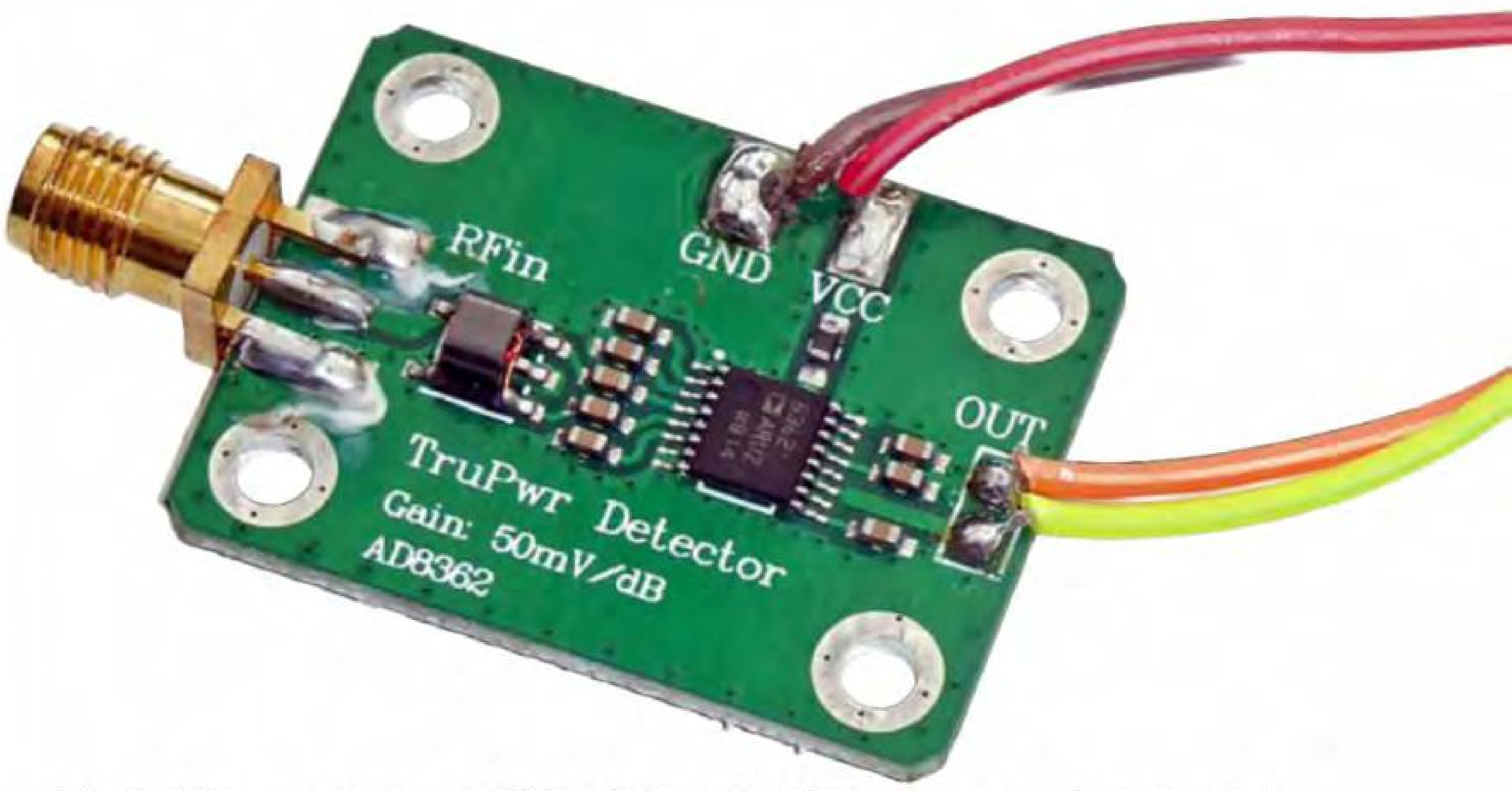


Fig.6.16: a low-cost AD8362 true RMS power-sensing module.

same relationship would apply if we had plotted dBm (dB relative to 1mW) against milliwatts (mW). Hence, 1mW = 0dBm, 10mW = 10dBm, 100mW = 20dBW and so on.

Fig.6.15 shows a typical analog power meter display with separate scales for RMS power and dBm. With its companion probe mount, this instrument is useful up to 40GHz.

True RF power meters

A similar arrangement to that shown earlier in Fig.6.11 can be used to indicate RF power, but in this case, the true RMS diode detector is in conjunction with a matched resistive load and the display scaled in milliwatts or watts. The load needs to be suitably rated in terms of power dissipation and must be purely resistive.

True power meters must also cope with the waveforms and crest factors (peak to average ratios) found in modern modulation systems such as quadrature amplitude modulation (QAM) and orthogonal frequency division multiplexing (OFDM). This places extra demands on the signal processing, requiring a fast response coupled with a wide measurement range, typically 60dB or more.

Typical of the devices currently available is the Analog Devices AD8362. This chip is a true-RMS-responding power detector offering a 65dB measurement range, designed for systems that demand an accurate response to true signal power. The chip operates from 50Hz to over 3.8GHz and accepts inputs from -52dBm to +8dBm.

The AD8362 requires a single +5V supply and consumes a mere 1.3mW when in the powered-down standby state.

A typical low-cost power sensing module based on an AD8362 is shown in Fig.6.16. The analog output of the module (0-5V) can be applied to a DC meter or to the analog port of a microcontroller like that shown in Fig.6.17, using code to display power in dBW or dBm if required, as shown on the LCD screen in that figure.

Note the use of a power attenuator in Fig.6.17 to reduce the signal level into the AD8362. The first line of the display in Fig.6.18 indicates the power supplied to the sensing module (which must be kept below

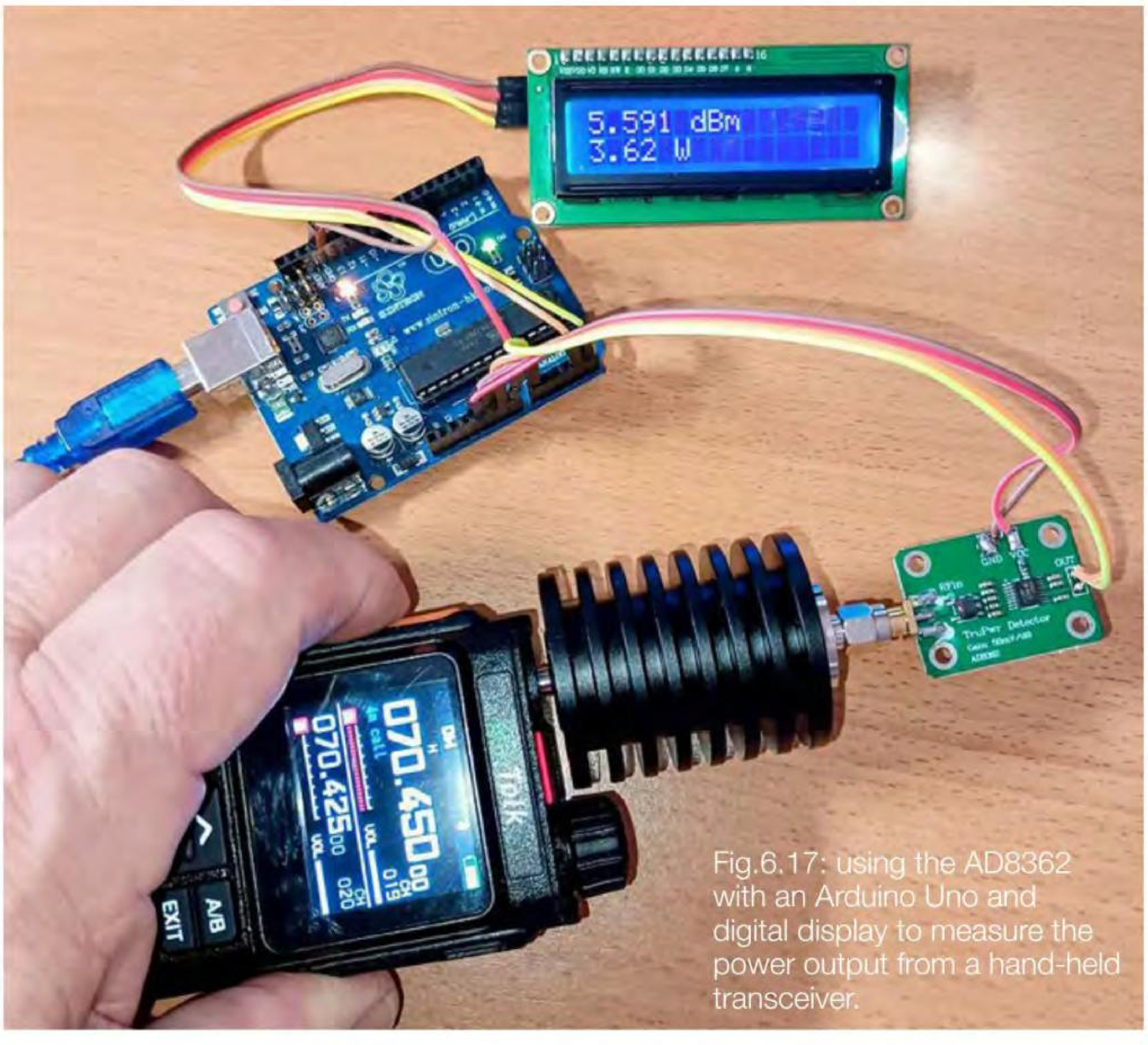


Fig.6.17: using the AD8362 with an Arduino Uno and digital display to measure the power output from a hand-held transceiver.

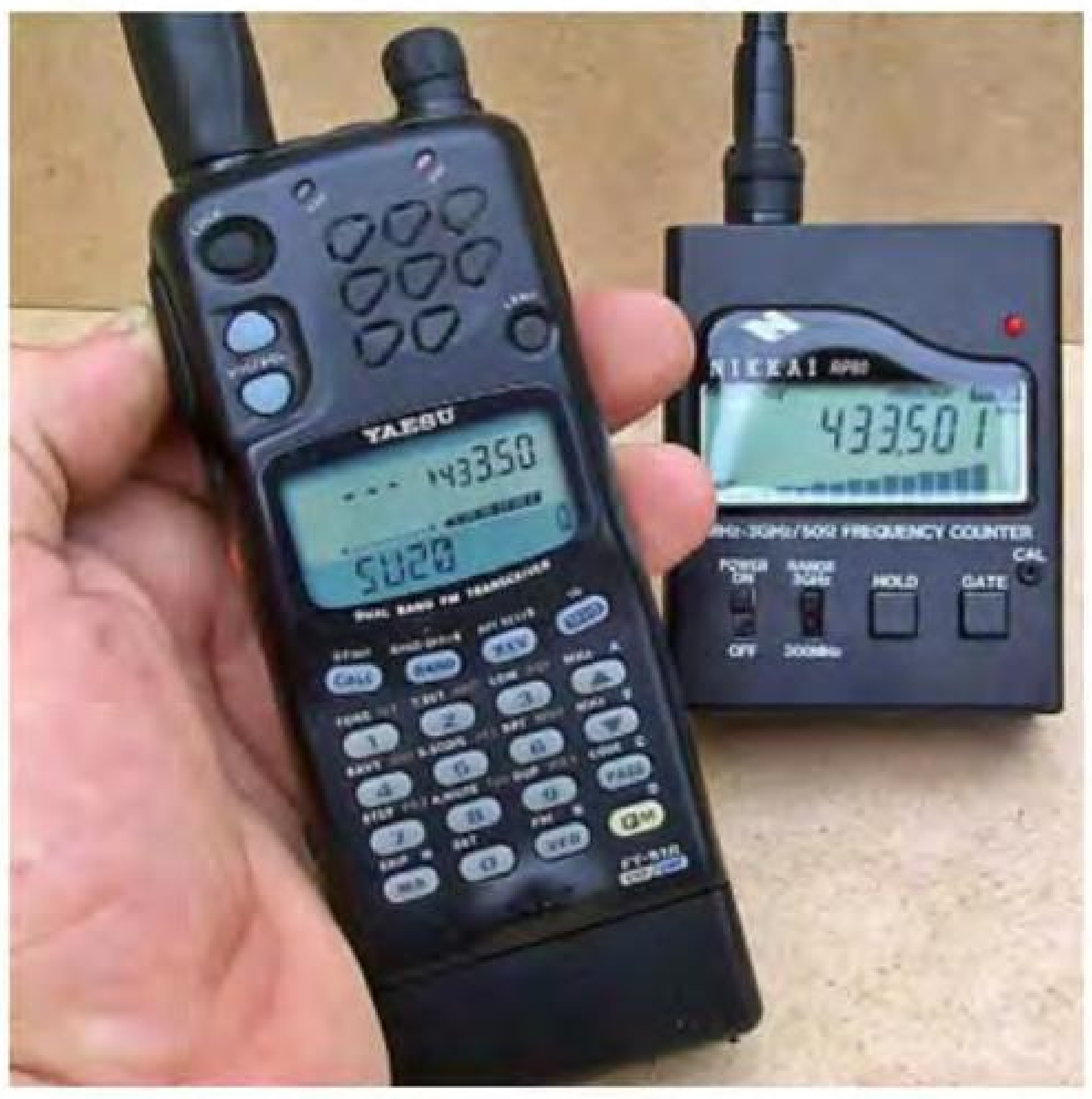


Fig.6.19: checking a dual-band FM transceiver. The DFM on the right indicates a frequency difference of 1kHz.

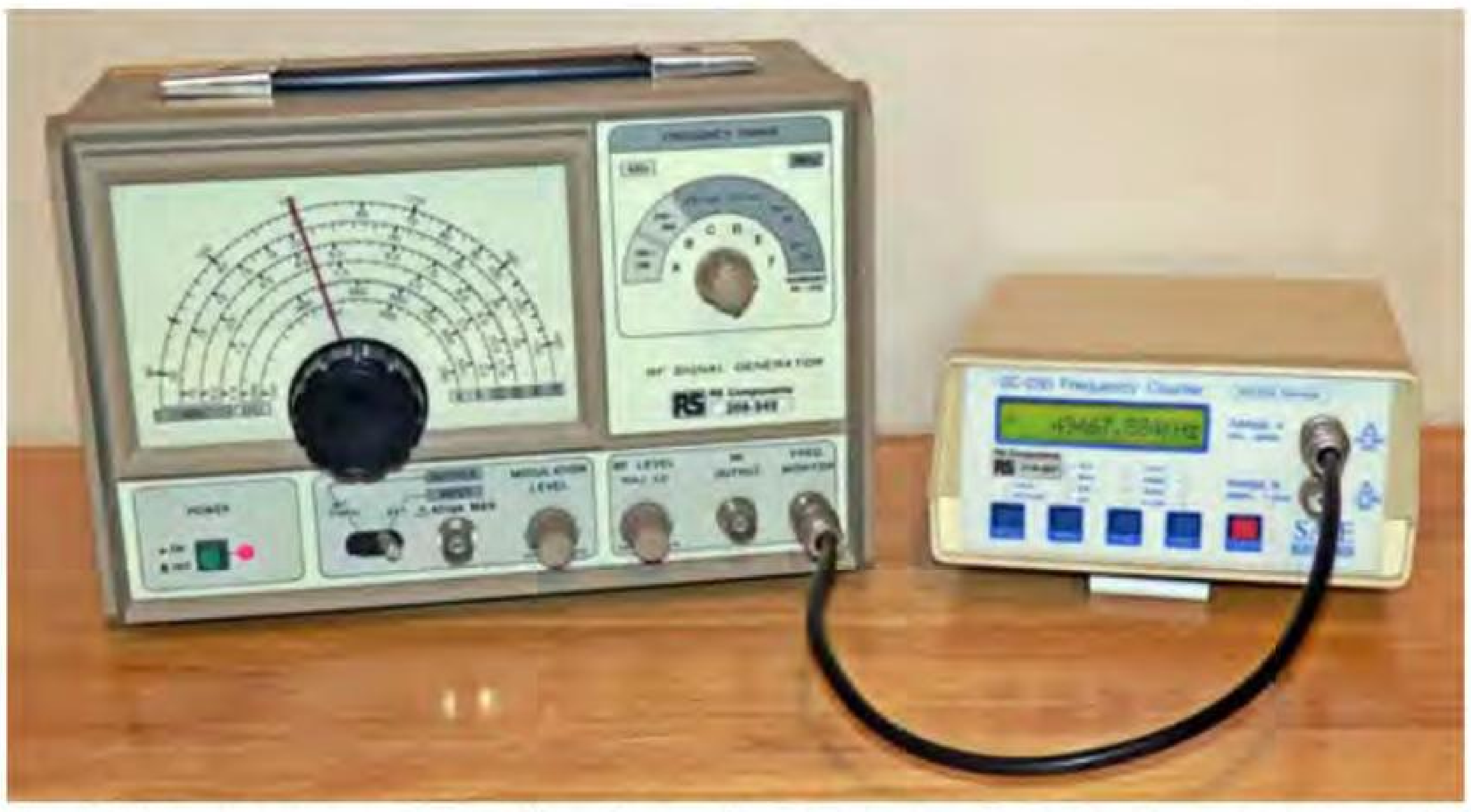


Fig.6.20: checking the calibration of an RF signal generator. The 8-digit DFM display shows 49467.584kHz with a 1Hz resolution.

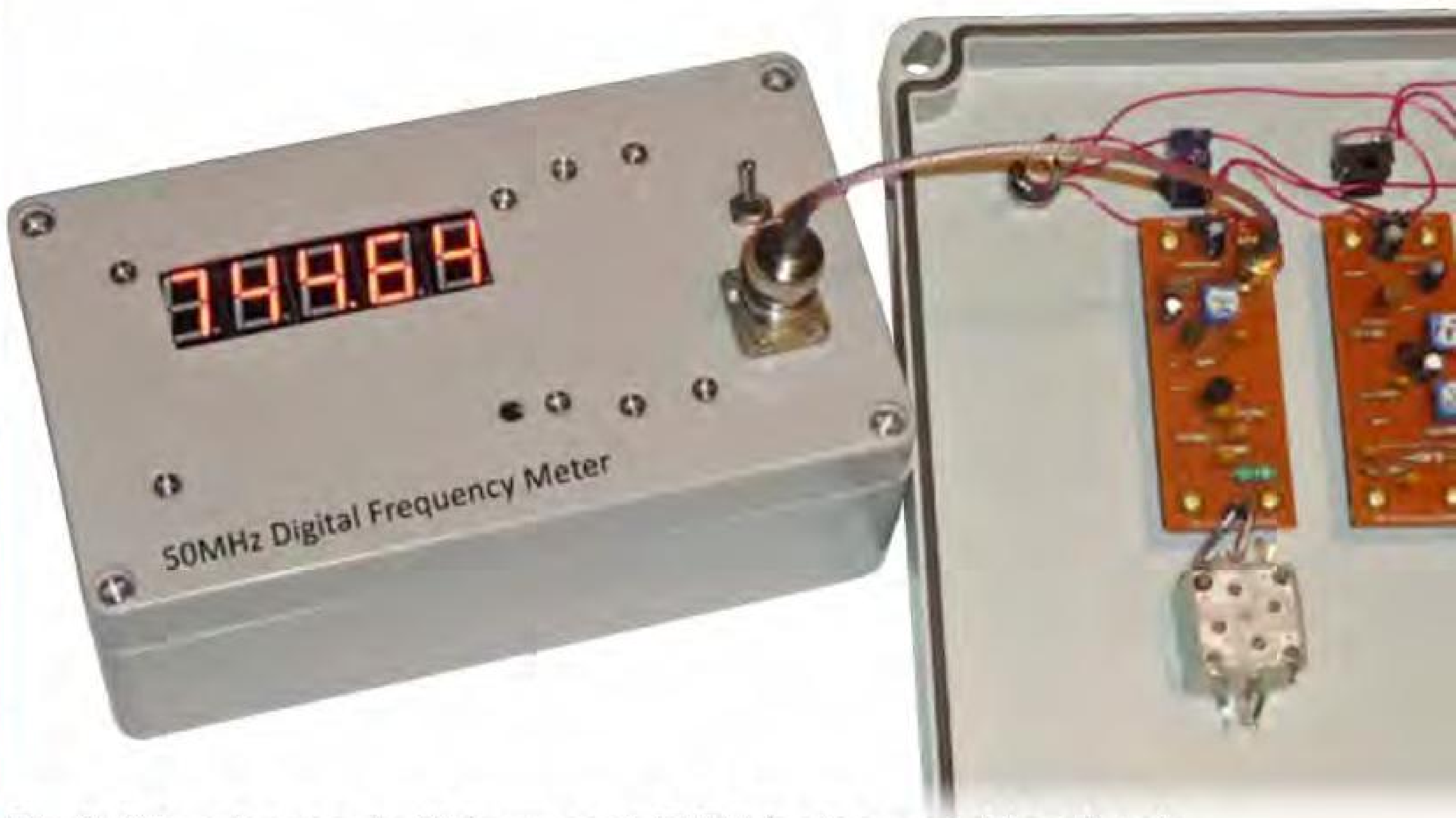


Fig.6.21: a home-built low-cost DFM being used to check the output frequency of the VFO described in Teach-In 12.3.

12dBm), while the second line indicates the power dissipated in the attached 50Ω load.

When using sensitive RF power measuring devices for indicating high levels of RF power, it's essential to use a suitably rated combined load and attenuator.

Measuring frequencies

Frequency is difficult to measure accurately using a conventional oscilloscope. Modern digital storage oscilloscopes (DSO) fare somewhat better in this respect, as they can often display frequency values digitally. However, a dedicated digital frequency meter (DFM) is often a better solution.

The desirable characteristics of such an instrument are high sensitivity and high upper frequency limit as well as adequate accuracy and resolution.

A sensitivity of 100mV$_{RMS}$ or better is suitable for most applications, and an upper frequency limit of 500MHz will be adequate for most measurements at HF and VHF. Most DFMs offer an accuracy of ten parts per million (0.001%). This is more than adequate for all but the most critical applications.

The resolution of an instrument depends on the number of digits in the display. For example, if a resolution of 100Hz is required when measuring a signal at 50MHz, a minimum of six display digits will be required.

The block diagram of a basic digital frequency meter is illustrated in Fig.6.18. The signal to be measured is initially sent through an input amplifier, followed by a Schmitt trigger buffer, which converts and shapes nearly all input signals into a logic-compatible pulse train. While the signal's waveform changes, the pulses still match the original input frequency.

Having a logic-compatible signal is important for connecting with the signal gate and the circuits that handle gating, counting, and latching.

The reference oscillator provides a very stable and accurate source for the time base circuit, typically using a temperature-controlled crystal oscillator (TCXO). TCXOs offer stability better than 1ppm, compared with standard quartz crystal oscillators, which usually have an error of about 30ppm, taking into account both initial accuracy, ageing, temperature drift etc.

The control pulses from the timebase have an accurate time duration and are applied to the signal gate which, in effect, opens and closes to allow a train of signal frequency pulses to pass through into the counter over a very precisely controlled interval.

These pulses are then electronically counted, and the result is passed to the display latches, which retain the count during the subsequent counting process.

To clarify this concept, consider a scenario in which a 25MHz signal is applied to the input of a five-digit DFM with its timebase set to 1ms. During this 1ms interval—when the gate remains open—25,000 pulses are counted, and the latch subsequently updates the display to indicate "25000kHz".

Conversely, if the time base is set to 1µs rather than 1ms, only 25 pulses are registered, resulting in a display of "00025MHz".

Figs.6.19 through 6.22 show examples of DFMs being used. In Fig.6.21, the five-digit display indicates a frequency of 744.64kHz with a resolution of 10Hz. Note how in Fig.6.22, an inline 30dB

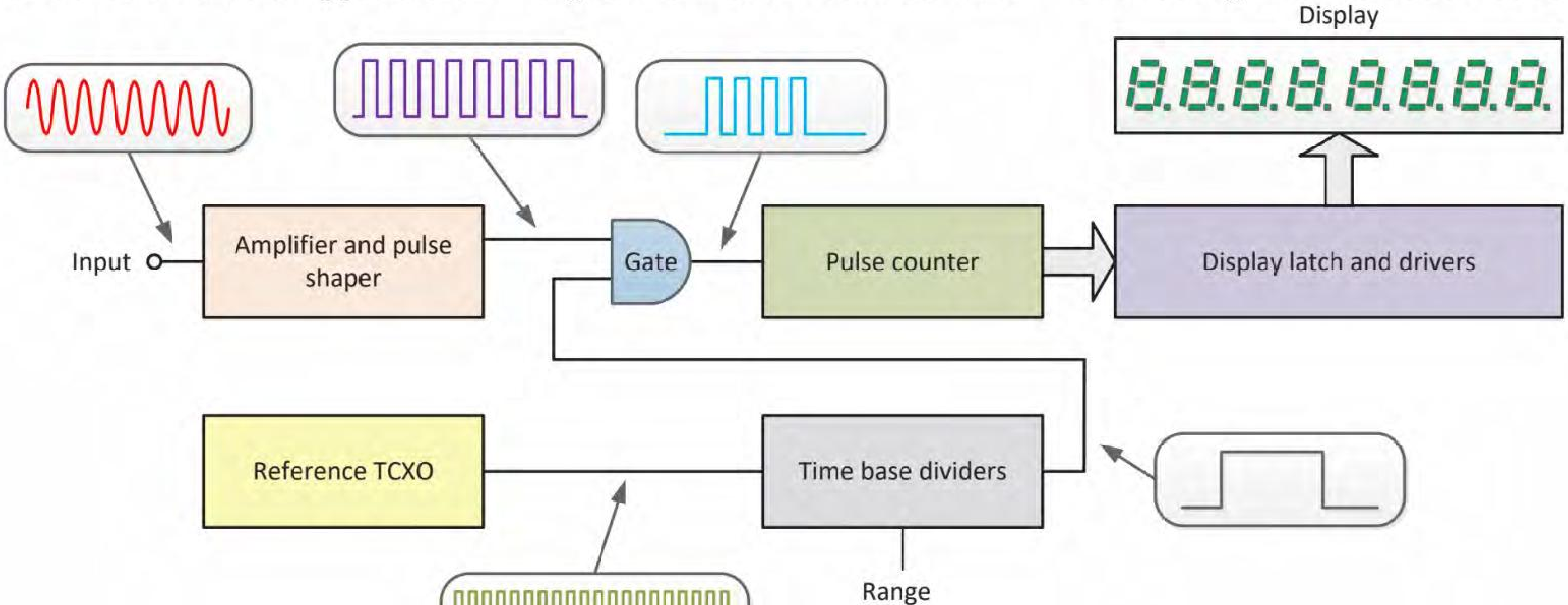


Fig.6.18: the simplified block schematic of a digital frequency meter (DFM).

power attenuator is connected between the transmitter output and DFM. This acts both as a matched load for the transmitter and as an attenuator to prevent overdriving the input of the DFM.

Hands-On: Handy RF voltmeter probe

This month's Hands-On project is a voltmeter probe that's ideal for signal tracing in RF circuits. The probe will allow you to quickly and easily make in-circuit measurements of RF voltages using only a low-cost digital meter for display. The probe can be calibrated for RMS (sinewave) voltages up to 20V over a frequency range from 25kHz to 95MHz.

The circuit of the RF voltmeter probe is shown in Fig.6.23. The probe uses two BAT42 Schottky diodes, D1 and D2, connected in a voltage-doubler arrangement like that shown previously in Fig.6.6. The DC output voltage (approximately twice the peak value of RF voltage applied to the probe tip) is developed across C2.

To overcome the non-linear diode characteristics, a small amount of bias voltage is applied to the detectors. This voltage can be adjusted by RV1 so that the two diodes are just on the point of conduction.

The bias supply is derived from a single 3V coin cell, which is mounted in a compact printed circuit board holder.

Lithium coin cells are a severe hazard to young children if they swallow them, so this board with the onboard cell holder must be secured in a plastic case held together with screws. Luckily, there is a neat case available that's the perfect size for the job (see the parts list for details).

As with our previous Hands-On projects, this one is constructed on a small piece of perforated copper stripboard measuring 25 × 64mm, arranged as nine strips each of 24 rows. The component layout (viewed from the top) is shown in Fig.6.24, while the corresponding track layout (viewed from below) is given in Fig.6.25.

The required 16 track breaks can be made using a spot face cutter or small drill bit, and the six links on the upper side of the board can be made using short lengths of tinned copper wire. Take care to observe the correct polarity of the two BAT42 diodes (the stripe on the package is the cathode connection).

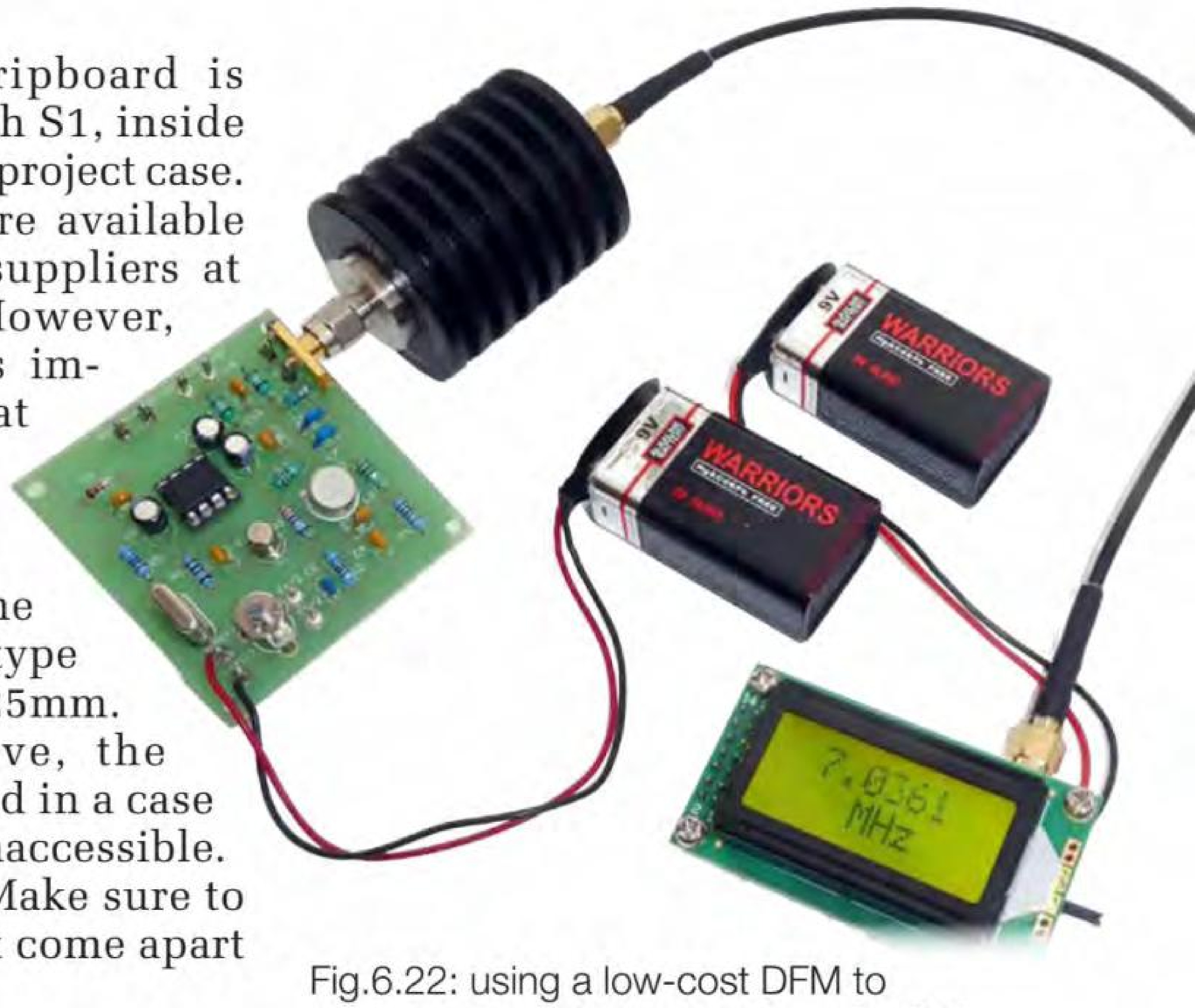


Fig.6.22: using a low-cost DFM to measure the output frequency of a 1W 7MHz transmitter.

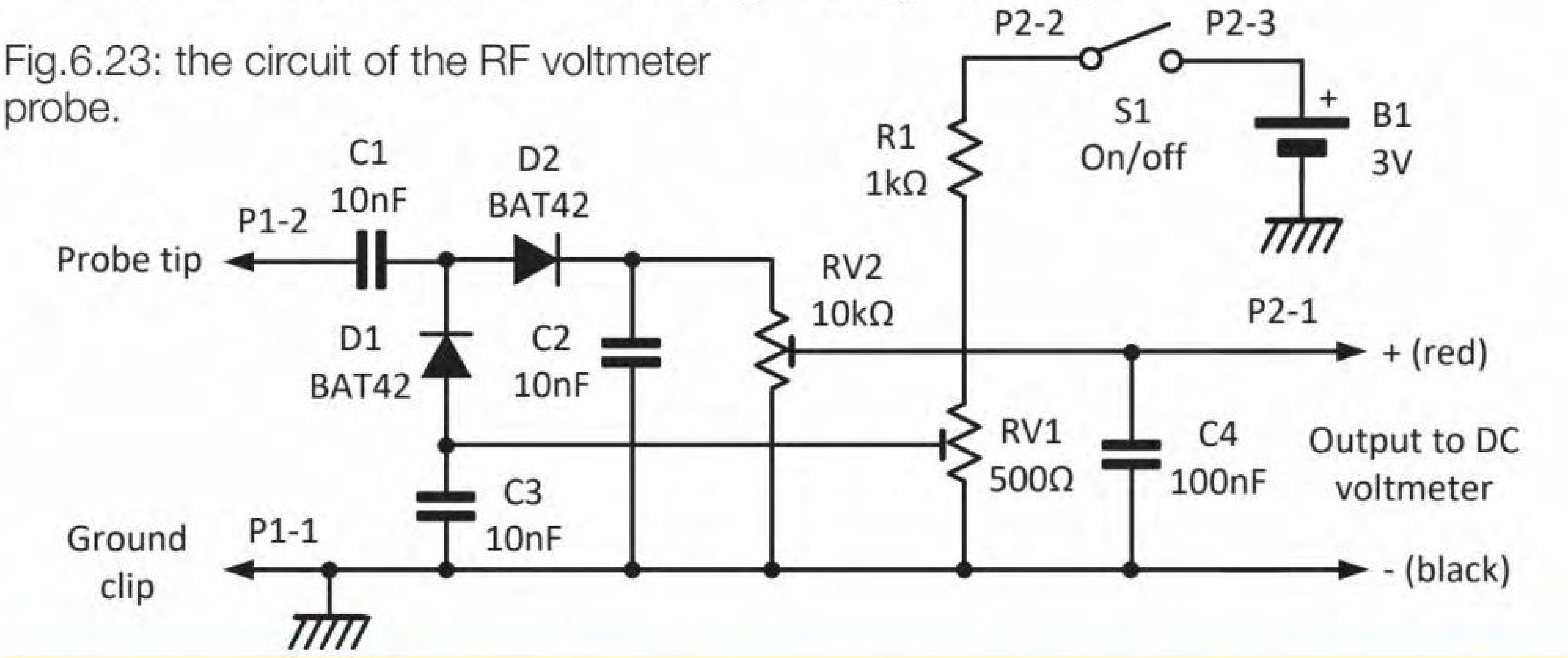
The completed stripboard is mounted, together with S1, inside an ABS probe or small project case. Suitable enclosures are available from several online suppliers at a reasonable cost. However, when purchasing, it's important to check that the dimensions of the case are sufficient to accommodate the board (25 × 64mm). The case used for the prototype measured 120 × 40 × 25mm.

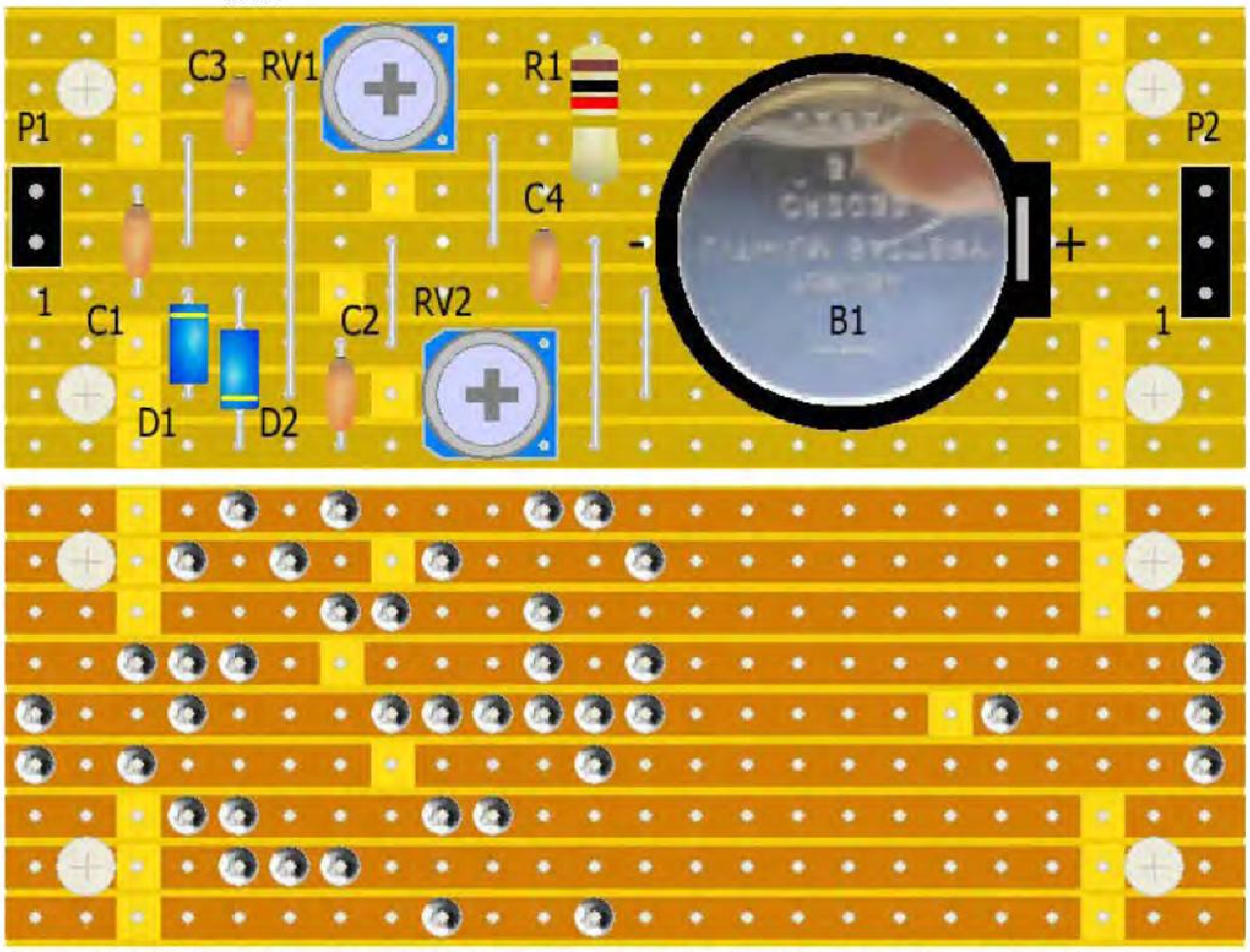
As explained above, the board must be mounted in a case to make the coin cell inaccessible. Don't skip that step! Make sure to use a case that doesn't come apart without tools.

Fig.6.26 shows the internal wiring of the RF voltmeter probe. The probe tip is connected to P1-2 using a short length of insulated hookup wire, and a 100mm length of insulated stranded wire (terminated with a crocodile clip) is connected to P1-1. This wire should be kept reasonably short to ensure good high-frequency performance.

The DPDT slide switch, S1, is connected to P2-2 and P2-3 using short lengths of insulated hook-up wire. 500mm lengths of black and red insulated multi-core flexible wire are used to link the DC output from the probe to the voltmeter. These leads should be fitted with 4mm banana plugs and the positive (red) and negative (black) connections terminated at P2-1 and P1-1, respectively.

Fig.6.23: the circuit of the RF voltmeter probe.





Figs.6.24 & 25: the component & track layouts for the RF voltmeter probe circuit board.

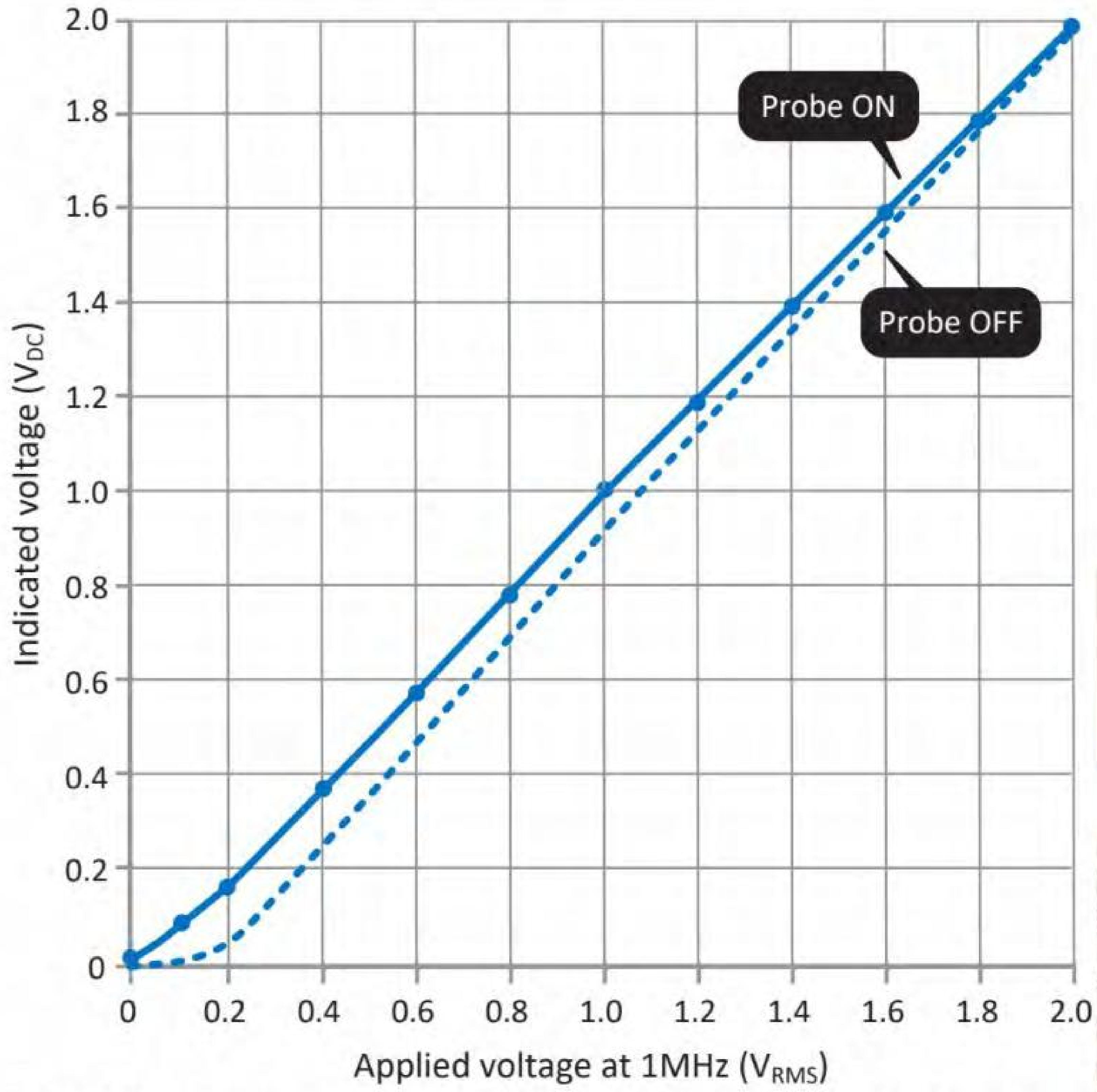


Fig.6.27: indicated voltage (VDC) plotted against input voltage (V_{RMS}) with the probe switched on and off.

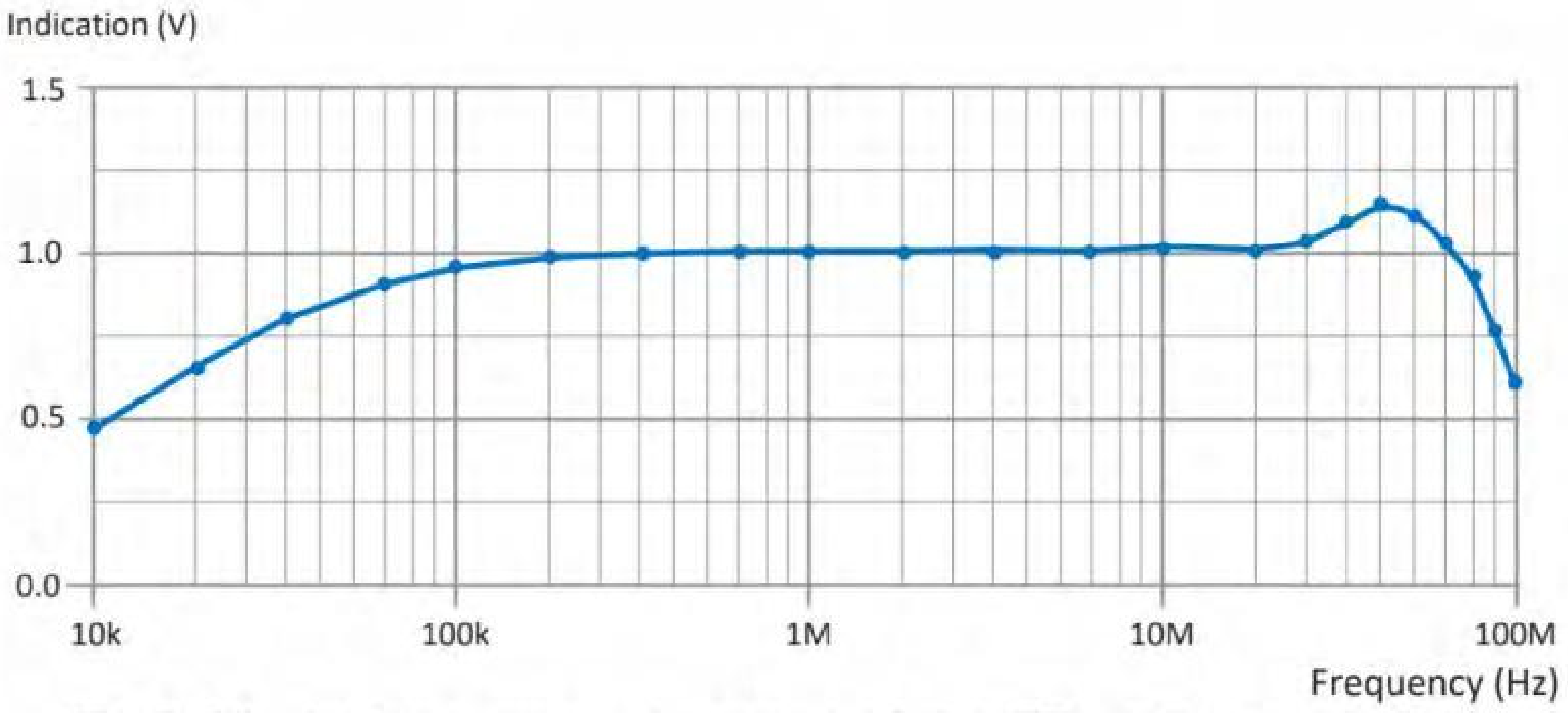


Fig.6.28: the frequency response of the RF voltmeter probe.

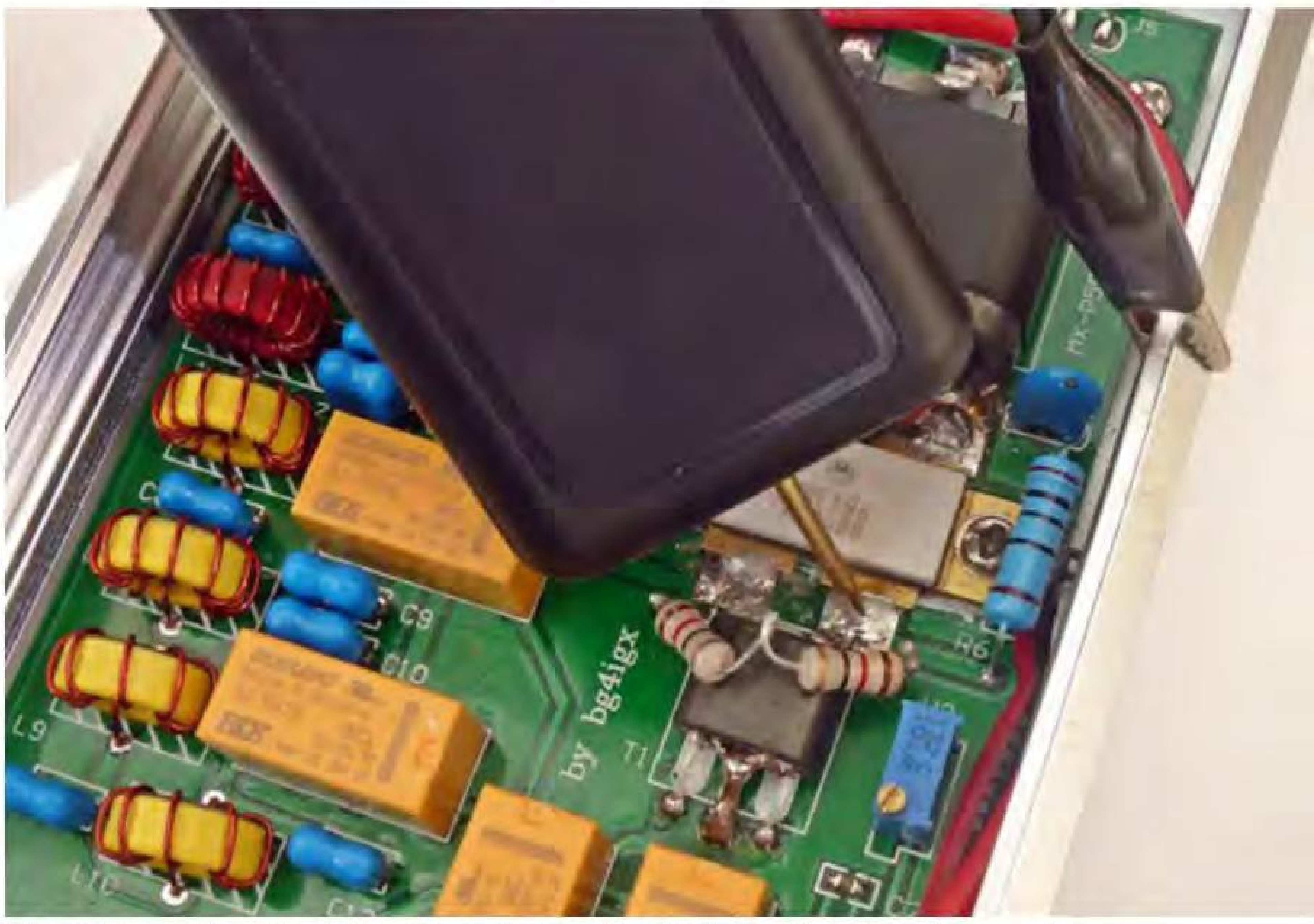


Fig.6.29: using the RF voltmeter probe to investigate voltages in an RF power amplifier.

Adjustment and calibration

The RF voltmeter probe requires initial adjustment and calibration using an RF signal source. If you don't have a suitable RF signal generator, you can use the VFO ('variable frequency oscillator') or reference source module that we described in Part 3 of Teach-In 12 (in the January 2026 issue). The following steps are required:

1. Switch the probe off and set both RV1 and RV2 on the probe to their fully anticlockwise positions. Insert the CR2032 3V coin cell into the holder with the positive (+) side up.
2. Leaving the probe tip and ground lead shorted together, connect the red and black output leads from the RF voltmeter probe to a digital voltmeter. Select the 20V DC range on the meter and keep it on this range for all subsequent steps.
3. Ensure that the probe tip remains unconnected and switch the RF voltmeter probe on. Ignore any reading on the meter at this stage.
4. Advance the setting on RV1 on the probe (turning it slowly clockwise) and adjust it for a reading of between 0.02V and 0.03V.
5. Connect the probe tip to SK1 on the VFO (or an alternative signal source of about 2V peak-to-peak at around 1MHz). Clip the ground connection to the 0V common rail.
6. Adjust the VFO to produce an output at about 800kHz and set RV1 on the VFO to mid-position. At this point, the meter should be indicating a reading of around 2V.
7. Note the reading on the meter and then adjust RV2 on the probe so that the meter reading falls to exactly 0.4 times (40% of) the initial reading indicated. For example, if the initial reading was 2.1V, RV2 should be adjusted so that the meter reads (2.1V × 0.4) = 0.84V (the RMS equivalent).
8. The RF probe should now indicate the approximate RMS value of the sinewave. If possible, the calibration should be checked against a known instrument and RV1 re-adjusted accordingly.

Fig.6.27 shows the response of the RF voltmeter probe to voltages below 2V. The result of applying a bias is evident, as is the error at applied voltages of less than 0.8V. For most purposes (and when carrying out relative adjustments), this small error can be ignored. Note the difference with the probe switched on and off.

The frequency response of the RF voltmeter probe is shown in Fig.6.28. The response is substantially flat from 100kHz to 10MHz, with a small resonant peak due to stray reactance at about 50MHz. Thereafter, the probe is useful to about 80MHz, above which the output falls rapidly.

Using the RF voltmeter probe

Fig.6.29 shows a typical application for the RF voltmeter probe when fault-finding on an HF power amplifier.

The probe tip is taken along the signal path to the points under investigation, and the ground clip left attached to a suitable ground point. In this case, the grounded metal enclosure has been used, but a nearby ground point on the PCB would also be suitable.

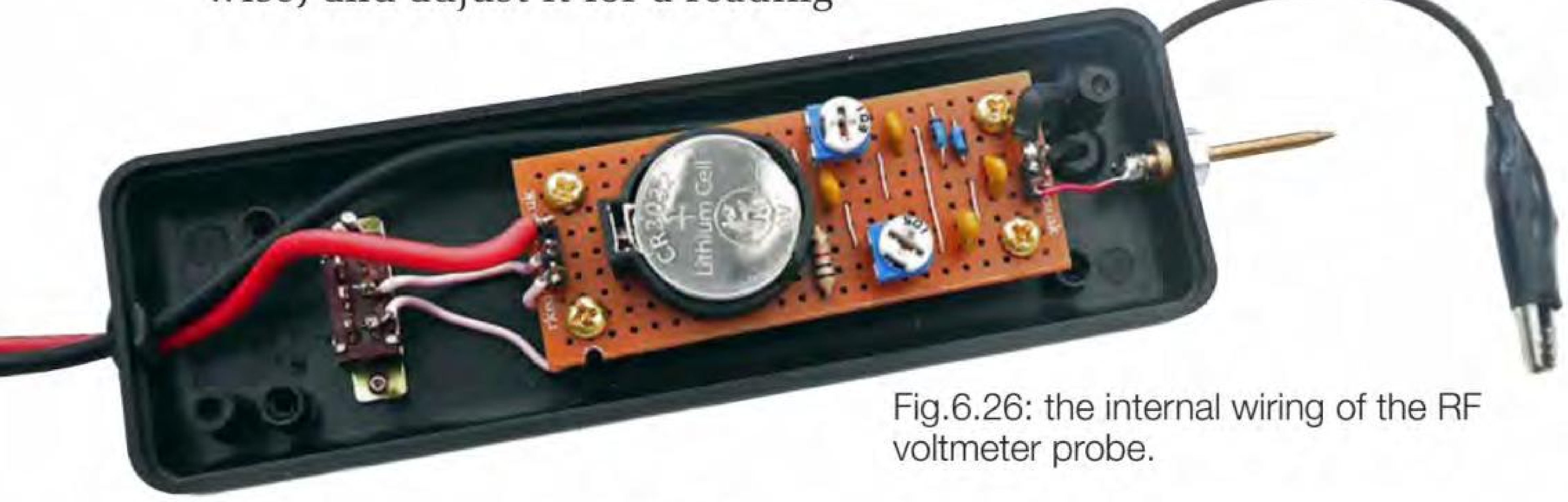


Fig.6.26: the internal wiring of the RF voltmeter probe.

Fig.6.30 shows how the RF probe can be used to check the operation of the amplitude modulator described in *Teach-In* Part 4 (February 2026 issue). Here again, the ground clip is attached to a convenient point on the common 0V rail and the probe tip moved to different test points. For comparison, we've shown the DC and RF test voltages at various points in the circuit.

The signal path can be clearly seen. It's also possible to determine the output power from the modulator, appearing across the 680Ω load resistor, R4. The 1.3VRMS measurement at this point (dropped across R4) can be used to determine the power dissipated in R4, as follows:

$$P_{out}=\frac{V_{RMS}^2}{R_L}=\frac{1.3V^2}{680\,\Omega}=0.0025\,\text{W} \quad \text{or} \quad 2.5\,\text{mW}$$

Coming up

In next month's instalment, we will be investigating low-cost spectrum and virtual network analysis. We will also have two Hands-On projects. One takes the form of a combined load and power meter that makes adjusting low-power HF transmitters a breeze; the other is a separate 25W load suitable for operation up to 450MHz. **PE**

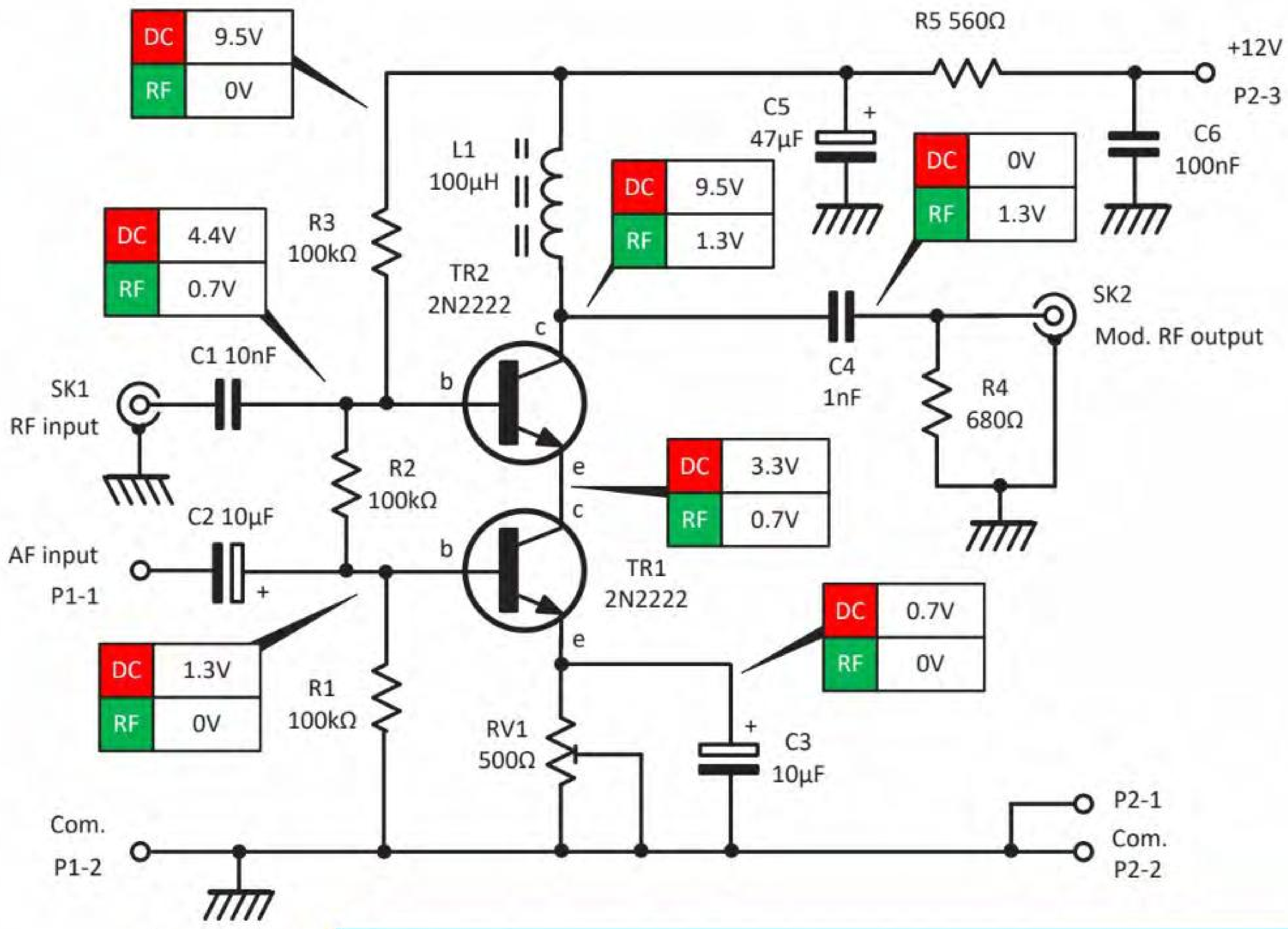


Fig.6.30: DC and RF voltage measurements made on the amplitude modulator circuit from *Teach-In* Part 4.

Listing 6.1: sample code for RMS scaling and dBV conversion.

```
void loop() {
  int rawValue = analogRead(inputPin);
  float voltage = (rawValue / ADC_STEPS) * V_REF;
  float voltRMS = voltage / 1.414;
  lcd.setCursor(0, 0);
  lcd.print(voltage/1.414, 3);
  lcd.print(" V");
  double dBV = 20.0 * log10(voltRMS);
  lcd.setCursor(0, 1);
  lcd.print(dBV, 2);
    lcd.print(" dBV");
  delay(1000);
  lcd.clear();
}
```

Parts List – RF voltmeter probe

1 25 × 64mm piece of stripboard (9 × 24 holes)
1 ABS probe case (129 × 40 × 25mm) [eBay 291019910762]
1 panel-mount probe (eg, machine screw sharpened to a point and matching nut)
1 2-pin male 0.1in/2.54mm header (P1)
1 3-pin male 0.1in/2.54mm header (P2)
1 DPDT miniature slide switch (S1)
1 PCB-mounting CR2032 coin cell holder (B1)
1 CR2032 3V lithium coin cell (B1)
4 brass or nylon hexagonal M3 × 10mm stand-off/spacer
4 M3 panhead machine screws
4 M3 countersunk head machined screws
1 red 4mm banana plug
1 black 4mm banana plug
1 insulated crocodile/alligator clip
1 50cm length of red PVC flexible test lead
1 50cm length of black PVC flexible test lead
1 10cm length of black PVC-insulated flexible wire

Semiconductors
2 BAT42 Schottky diodes (D1, D2)

Capacitors
1 100nF 50V ceramic (C4)
3 10nF 50V ceramic (C1-C3)

Resistors
1 1kΩ ¼W axial, 5% or better (R1)
1 500Ω top-adjust miniature preset variable/trimpot (RV1)
1 10kΩ top-adjust miniature preset variable/trimpot (RV2)

ENIGMA
COMPONENT SHOP
www.enigma-shop.com
Electronic Components for the Hobbyist!
Cable & Wire • Capacitors • Connectors • Filters
Kits & Modules • Hardware & Cooling
and much more...
READY TO BE REWARDED?
Join our loyalty scheme and **earn points** on every purchase you make which can be **used for discounts** in the future.
Read our reviews on Trustpilot

DOWNLOAD YOUR PE/EPE BACK ISSUES

COMPLETE-YEAR collections – all 12 issues for each year!
From just £5.95 each

VOL 28 – January to December 1999 – **£5.95**
VOL 29 – January to December 2000 – **£6.95**
VOL 30 – January to December 2001 – **£7.95**
VOL 31 – January to December 2002 – **£7.95**
VOL 32 – January to December 2003 – **£7.95**
VOL 33 – January to December 2004 – **£7.95**
VOL 34 – January to December 2005 – **£8.95**
VOL 35 – January to December 2006 – **£8.95**
VOL 36 – January to December 2007 – **£8.95**
VOL 37 – January to December 2008 – **£8.95**
VOL 38 – January to December 2009 – **£10.95**
VOL 39 – January to December 2010 – **£11.95**
VOL 40 – January to December 2011 – **£12.95**
VOL 41 – January to December 2012 – **£14.95**
VOL 42 – January to December 2013 – **£15.95**
VOL 43 – January to December 2014 – **£16.95**
VOL 44 – January to December 2015 – **£17.95**
VOL 45 – January to December 2016 – **£19.95**
VOL 46 – January to December 2017 – **£22.95**
VOL 47 – January to December 2018 – **£27.95**
VOL 48 – January to December 2019 – **£27.95**
VOL 49 – January to December 2020 – **£29.95**
VOL 50 – January to December 2021 – **£32.95**
VOL 51 – January to December 2022 – **£34.95**
VOL 52 – January to December 2023 – **£34.95 – NEW!**
VOL 53 – January to December 2024 – **£34.95 – NEW!**

FIVE-YEAR collections – 60 issues!
From just £14.95 each

PE2832: Jan 99 to Dec 03 – £14.95
PE2933: Jan 00 to Dec 04 – £14.95
PE3034: Jan 01 to Dec 05 – £14.95
PE3135: Jan 02 to Dec 06 – £14.95
PE3236: Jan 03 to Dec 07 – £14.95
PE3337: Jan 04 to Dec 08 – £14.95
PE3438: Jan 05 to Dec 09 – £19.95
PE3539: Jan 06 to Dec 10 – £25.95
PE3640: Jan 07 to Dec 11 – £25.95
PE3741: Jan 08 to Dec 12 – £25.95
PE3842: Jan 09 to Dec 13 – £33.95
PE3943: Jan 10 to Dec 14 – £33.95
PE4044: Jan 11 to Dec 15 – £33.95
PE4145: Jan 12 to Dec 16 – £35.95
PE4246: Jan 13 to Dec 17 – £37.95
PE4347: Jan 14 to Dec 18 – £38.95
PE4448: Jan 15 to Dec 19 – £39.95
PE4549: Jan 16 to Dec 20 – £42.95
PE4650: Jan 17 to Dec 21 – £44.95
PE4751: Jan 18 to Dec 22 – £44.95 – NEW!
PE4852: Jan 19 to Dec 23 – £44.95 – NEW!

20 YEARS OF PE!

AMAZING BUNDLE OFFER
Every issue from 1999 – 2018

240 magazines for just

£69.95

PE2832: Jan 99 to Dec 03
PLUS!
PE3337: Jan 04 to Dec 08
PLUS!
PE3842: Jan 09 to Dec 13
PLUS!
PE3447: Jan 14 to Dec 18

Download your collections at: www.electronpublishing.com

Constructional Project

BAT+
BAT–
TEST BATT 8.6V
BATTERY: 0.0V
MAX CURR: 1A
V Λ > M
POWER
DOWN
UP
ENTER
MODE
SILICON CHIP
Versatile Battery Checker

VERSATILE BATTERY CHECKER

This versatile tool allows you to check the condition of all manner of batteries, including Li-ion, LiPo, lead-acid, SLA, AGM, alkaline and zinc-carbon types. It can test 12V or 24V vehicle starter batteries as well as smaller 9V batteries and cells with a much lower voltage, including D, C, AA and AAA cells.

PROJECT BY TIM BLYTHMAN

Voltmeters (such as the ones built into multimeters) are a simple way to check the terminal voltage of a battery and can help to estimate its state of charge.

However, voltmeters have a high input impedance, so they do not expose the battery to any significant load. Thus, a voltmeter reading does not indicate a battery's internal resistance.

Increasing internal resistance can be a sign of degradation and loss of capacity in a battery. We've seen batteries with a perfectly reasonable terminal voltage that completely 'throw in the towel' when exposed to any kind of load! Such a battery cannot be relied upon. So you really need a proper battery checker, like this one.

A good battery tester will apply a pulsed load to the battery over a range of possible currents, from a few amps to many tens of amps. This can be adjusted to suit the size of the battery. A healthy vehicle starting battery, for example, won't show any significant drop in terminal voltage unless you are drawing many amperes, and it's best to keep that brief to protect both the battery and the device testing it.

We have seen simple battery tester circuits in the past that used discrete logic ICs to control several Mosfets wired in parallel, with a meter arranged so it would show a reading relative to the momentary battery terminal voltage drop due to that load.

Nowadays, we can use a modern microcontroller with an ADC (analog-to-digital converter) to control the sequencing of such a device. Its ADC can measure voltage and perform calculations to display results in an easy-to-read text form. That makes for a much more compact instrument than some of the other designs we've seen.

It also allows us to test batteries below 6V, such as the now very common ~3.7V lithium-ion, LiPo & $LiFePO_4$ cells. It would also be handy to be able to test AA, AAA, C and D cells and such, as well as batteries made from them. That isn't possible with self-powered designs, since these cells do not provide enough voltage to run circuitry, so we have added the option of a separate battery to power the Checker.

The Versatile Battery Checker is easy to use. You can set a voltage drop limit (specified in percent; we have set the default to 10%) and dial in the maximum test current. The test sequence starts at the press of a button, and the results are reported in about one second.

The Checker runs 10 test pulses spread up to the maximum current limit. If at any time the test current is exceeded or the battery voltage drops too much, the remaining tests are cut short and the results of the completed pulses are reported. The Checker also monitors for conditions that might otherwise damage the hardware and cancels pending tests in such cases.

Our Versatile Battery Checker

Perhaps the best way to explain our new design is to examine the circuit diagram, Fig.1. The battery under test (BUT) connects between two binding posts, CON3 (positive) and CON4 (negative). The path for the test current is through diode D1, Mosfet Q1 and a 15mΩ current-measuring shunt.

In the absence of any other signals, Q1 is held off by the 100kΩ resistor

Versatile Battery Checker

connected to its gate via a 220Ω resistor. Diode D1 is for reverse-polarity protection, since Q1 would otherwise conduct excessive reverse current through its body diode if the connections were reversed.

The circuit can run off the BUT, receiving power to its main V+ rail via diode D2. A 9V battery connected at CON2 can also supply power. Q5 is a PNP transistor arranged as a high-side switch that can source power to V+ via diode D3.

From V+, PNP transistors Q2 and Q7 form a 600μA current-limited source that can be enabled by applying current to NPN transistor Q3's base. This 600μA flows out of Q2's collector and into Q1's gate, tending to bias it on. A current source is used here so that the circuit's operation is consistent even if the V+ voltage varies (and it likely will if running off the BUT).

The section around NPN transistor Q4 provides the current control function. Assume for now that the line labelled CURCON is connected to circuit ground. As the current through the 15mΩ shunt rises, so does the voltage at Q4's base.

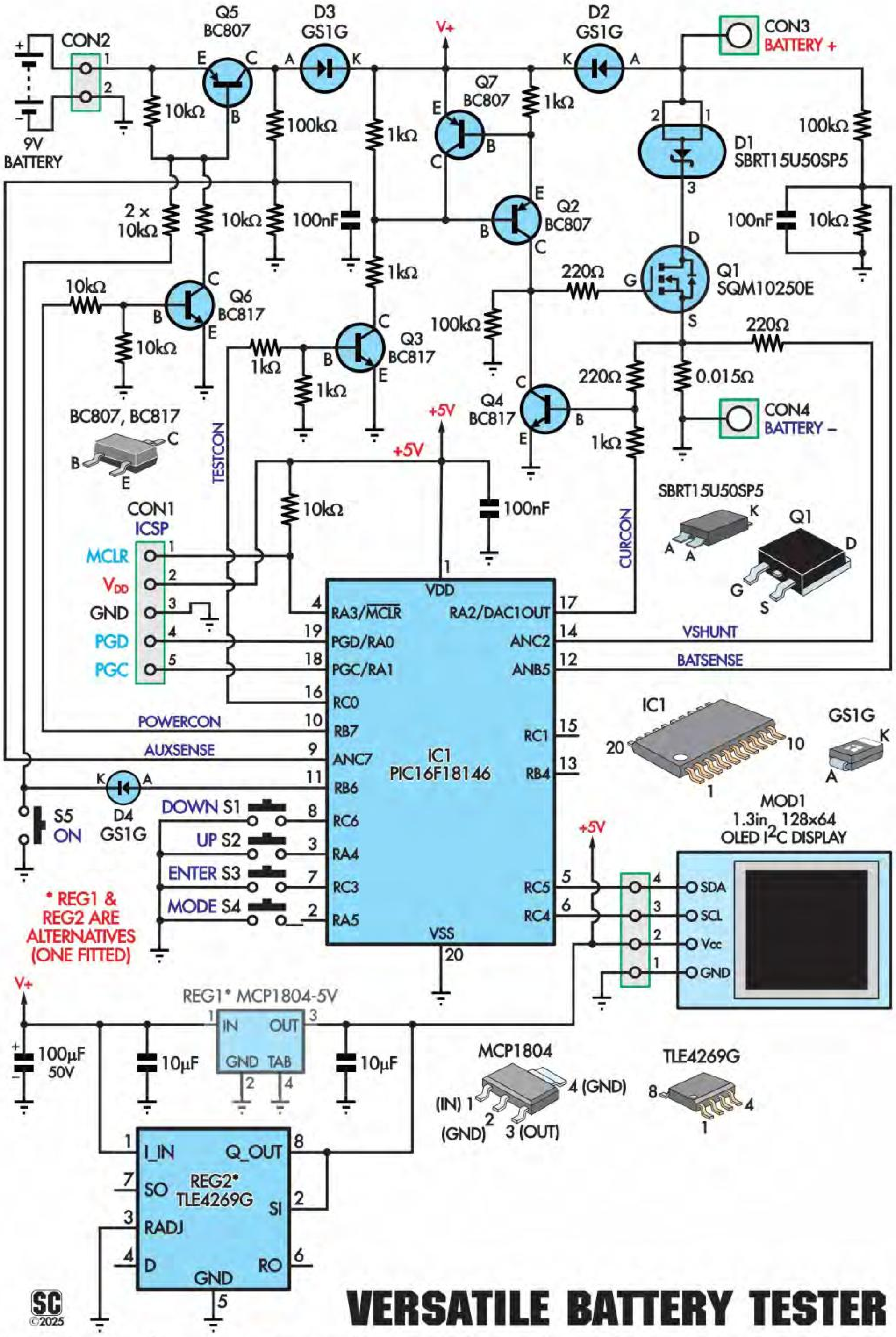
When the voltage across the shunt reaches about 0.73V, the divider can supply 0.6V to Q4's base. This will switch on Q4 and shunt the current from Q2 away from Q1's gate, reducing its gate bias voltage, and maintaining the current at a level that keeps this state. To achieve this, a nominal 48A needs to flow through the shunt.

If we apply 3.3V to the CURCON line then, even if no current is flowing, Q4 has 0.6V at its base and the Mosfet is forced off. Between these two extremes, we can set a voltage that will approximately set the current that is flowing through the shunt and thus flowing out of the BUT.

Of course, the voltage at Q4's base will not strictly be 0.6V, and there are some variations in the other voltages, but the basic principle remains valid. Later, we'll look at how this voltage is set. Since the shunt is on the low side (BAT−) of the circuit, the voltage developed across it (relative to circuit ground) is proportional to the actual current flowing, and the microcontroller can easily measure that.

Power supply

The control circuitry runs at a nominal 5V supplied from either REG1 or REG2. Only one of these regulators should be fitted; the two parts are simply alternatives that perform the same role.



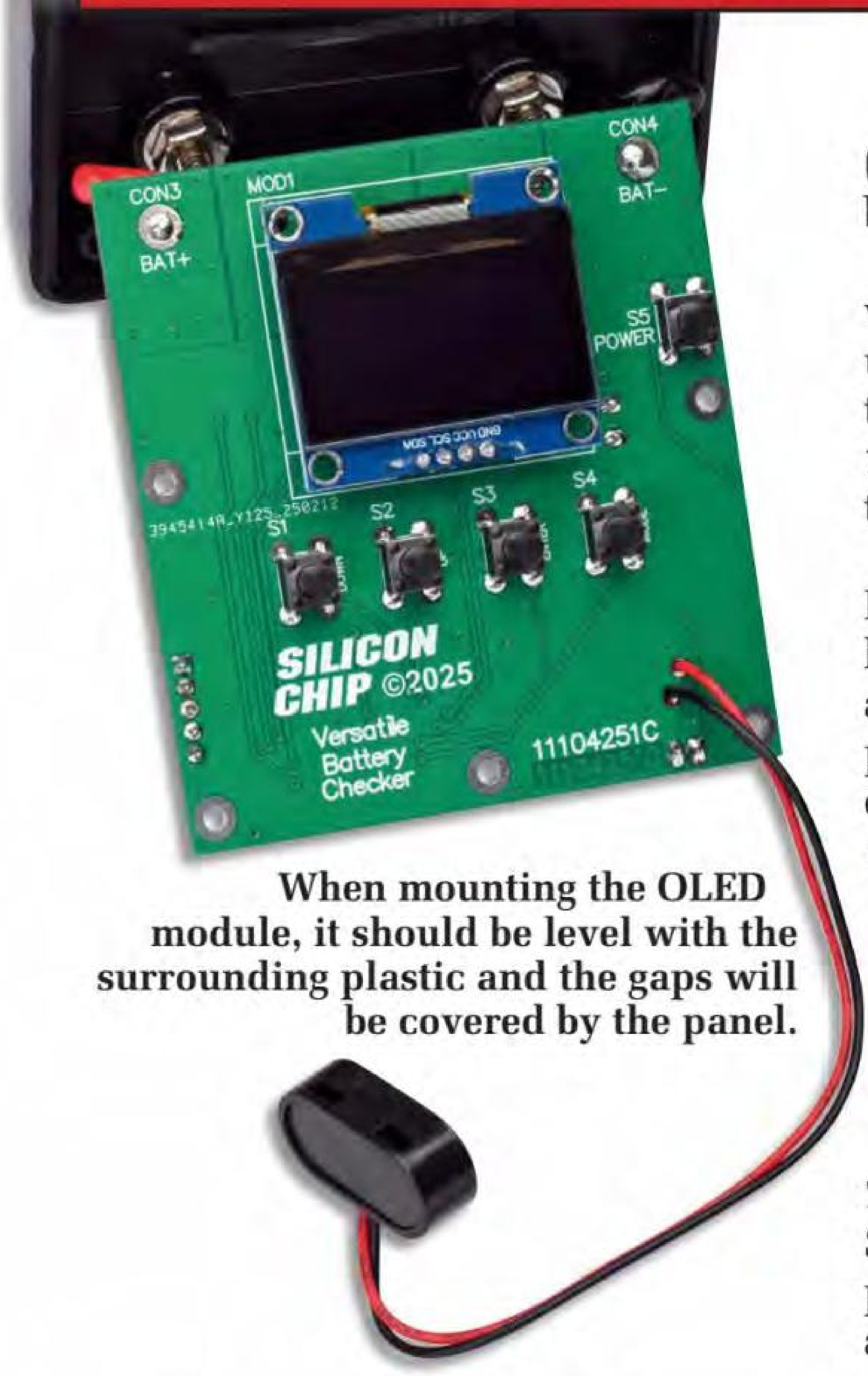
CON2
9V BATTERY
Q5 BC807
D3 GS1G
V+
D2 GS1G
CON3 BATTERY +
Q7 BC807
Q2 BC807
D1 SBRT15U50SP5
Q1 SQM10250E
Q6 BC817
Q3 BC817
Q4 BC817
2 × 10kΩ
10kΩ
100kΩ
100nF
1kΩ
220Ω
0.015Ω
CON4 BATTERY −
BC807, BC817
TESTCON
CURCON
+5V
CON1 ICSP
MCLR
VDD
GND
PGD
PGC
IC1 PIC16F18146
VDD
RA3/MCLR
PGD/RA0
PGC/RA1
RC0
RB7
ANC7
RB6
RC6
RA4
RC3
RA5
VSS
RA2/DAC1OUT
ANC2
ANB5
RC1
RB4
RC5
RC4
VSHUNT
BATSENSE
POWERCON
AUXSENSE
SBRT15U50SP5
GS1G
S5 ON
D4 GS1G
DOWN S1
UP S2
ENTER S3
MODE S4
* REG1 & REG2 ARE ALTERNATIVES (ONE FITTED)
MOD1 1.3in 128x64 OLED I2C DISPLAY
SDA
SCL
Vcc
GND
REG1* MCP1804-5V
IN OUT GND TAB
100μF 50V
10μF
MCP1804
(IN) 1
(GND) 2
3 (OUT)
4 (GND)
TLE4269G
REG2* TLE4269G
I_IN
Q_OUT
SO
SI
RADJ
D
RO
GND
SC ©2025
VERSATILE BATTERY TESTER

Fig.1: key to this circuit's operation is Mosfet Q1 being driven in constant-current mode with the target current set by the voltage on the CURCON line, produced by IC1's internal DAC. The circuitry at upper left provides pushbutton power control using S5 and Q5. Two 100kΩ/10kΩ dividers allow the internal and external battery voltages to be sensed.

The TLE4269G (REG2) can handle an input voltage up to 45V. We got a fairly large number of these nice chips inexpensively, so will supply them in kits. In case it becomes hard to find, an MCP1804 (REG1) can be used instead. This can handle up to 28V; that isn't high enough to comfortably run from a fully charged 24V battery, which can reach nearly 30V.

We'll assume REG2 is fitted, since that is what we used on our prototype. It comes in the SOIC-8 package, with features not available on the 3-pin MCP1804 (the latter's tab is connected to the middle pin).

The connections to pins 2 and 3 of REG2 simply disable its extra features and pins 1, 5 and 8 provide the minimum input, ground and output connections needed.



When mounting the OLED module, it should be level with the surrounding plastic and the gaps will be covered by the panel.

The two 10µF capacitors provide the necessary bypassing required by either regulator. Diodes D2 and D3, noted earlier, allow REG2 to be powered from either of the two sources. There is also a 100µF capacitor that holds up the V+ line during tests. This is important if the BUT is used to provide the supply current.

Microcontroller

IC1 is a PIC16F18146 8-bit microcontroller and it has a 100nF bypass capacitor fitted to its supply at pins 1 and 20 (ground). We make use of several of its internal peripherals. Importantly, it has an internal voltage reference that can be fed to an 8-bit DAC (digital-to-analog converter) with a buffered output at pin 17.

The DAC is used to set the CURCON voltage and thus the BUT current. We use the 4.096V internal reference, so the DAC has an output resolution of 16mV, which maps to steps of ¼A for the BUT.

The microcontroller connects to ICSP (in-circuit serial programming) header CON1 with its 5V supply rails, along with pins 4, 18 and 19. Pin 4 is pulled up to 5V to prevent inadvertent resets. We used CON1 for development, but it is does not need to be fitted unless IC1 needs to be programmed in-circuit.

The remaining pins on IC1 are general-purpose I/O pins (GPIOs) used for straightforward digital and analog input functions. Pins 2, 3, 7 and 8 connect to tactile switches S1-S4. These pins have an internal pullup current enabled, so they sit at a high level unless the switch is pressed, pulling it to ground and causing the digital input to change state.

Pins 5 and 6 connect to OLED module MOD1, providing an I^2C serial interface, along with the 5V supply rails. The switches and display form the user interface; we'll delve into its details a bit later.

Pins 9 and 12 connect to identical 100kΩ/10kΩ dividers supplemented by 100nF capacitors on their lower legs. These are used with IC1's ADC peripheral to monitor the voltage at the 9V battery at CON2 and the BUT, respectively.

The internal 4.096V internal reference is used for these measurements, giving a range of around 45V with a resolution of 11mV using the 12-bit ADC.

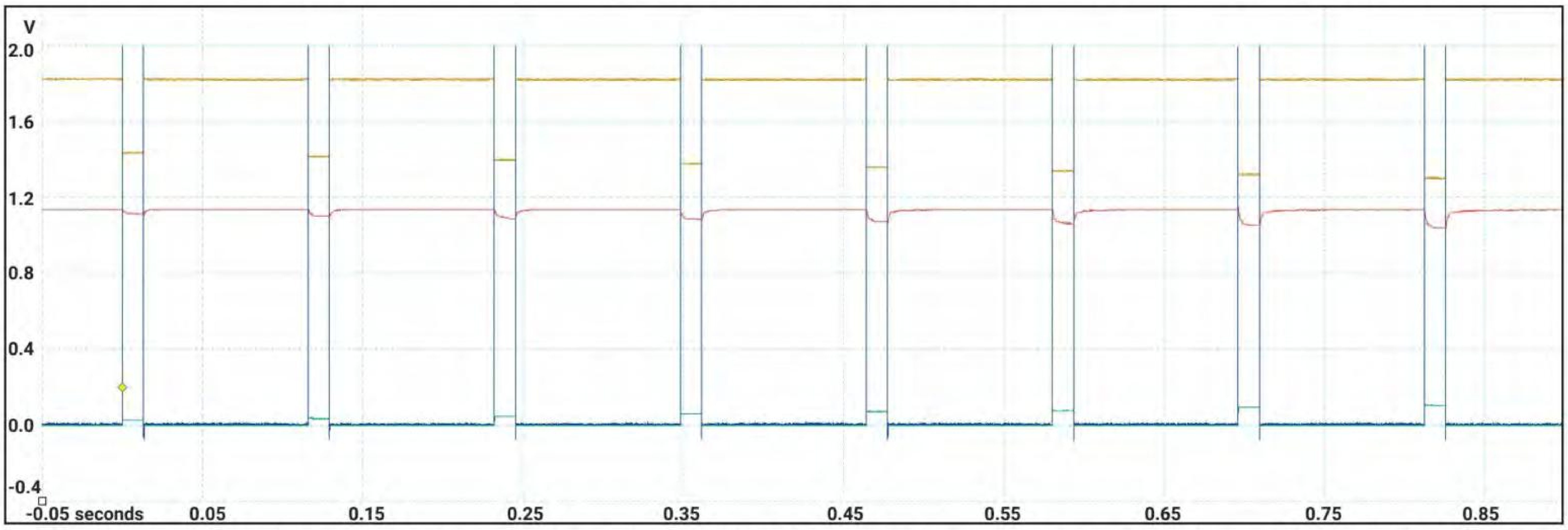
Pin 14 is similarly used to monitor the voltage at the shunt and thus measure the current drawn by the BUT. The current measurements use a 1.024V reference, allowing currents up to 65A to be measured. The measured value of these internal references is written into the chip at manufacture, so we can use them without an extra calibration step.

Measuring the change in BUT voltage due to various current loads is the essence of what the Checker does. These measurements also allow, for example, an internal resistance value to be calculated.

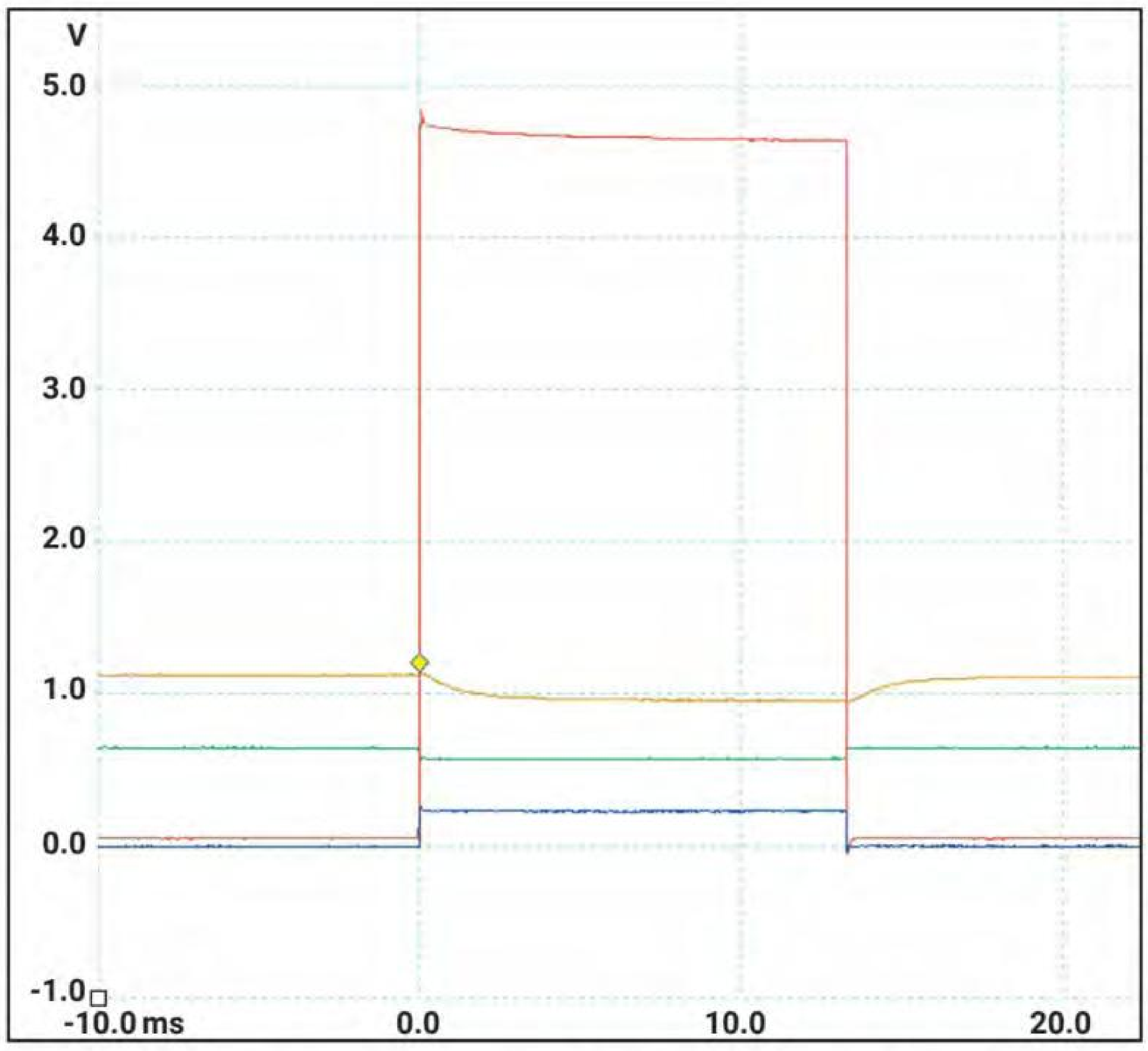
We mentioned Q5 earlier, but not how it is controlled. Q5 can be switched on either by closing S5 or by raising the voltage on the POWERCON line (IC1's pin 10), which switches on Q6. A typical sequence might involve pressing S5, which powers on the microcontroller. The micro then biases on Q6 to maintain power, and the button can be released.

The micro can then switch itself off later by pulling POWERCON low, to 0V. This might be done under user control or after a timeout. The micro applies a pullup current to pin 11, allowing it to detect when S5 is pressed. D4 is used to prevent voltages above 5V from feeding back into the microcontroller, which could damage it.

Pin 16, the TESTCON line, can be taken high to switch on Q3, which in turn activates the Q2 current source. This gives us two ways to ensure that Q1 is switched off between tests, since we can also put up to 4V on CURCON,



Scope 1: eight pulses from a test sequence. Blue is TESTCON (which, at 5V peak, has exceeded the scale), red is the scaled battery voltage (BATSENSE), green is current (VSHUNT) and yellow is CURCON, offset for clarity (the peak level is nominally 4V). As CURCON drops, the VSHUNT curve indicates an increasing current and the battery voltage drops further.



Scope 3: this is like Scope 2 but with a closeup of a single pulse, except yellow is now the scaled battery voltage (BATSENSE). Note how Q1's gate drive adjusts as the battery voltage sags. The firmware allows 10ms for the voltages to stabilise before taking several samples over a few more milliseconds. All the important voltages settle before sampling.

forcing Q4 on and keeping Q1's gate low.

Software overview

The user interface for the Versatile Battery Checker is quite simple since there is not much to configure between tests. There is a single page that controls the test process. Initially, it shows the connected BUT voltage, and the buttons allows the test current to be set and the test started.

Unlike some other battery checker designs we have seen, which only run a handful of tests at a single current setting, the Versatile Battery Checker runs up to 10 tests spread out from near zero up to a maximum target current.

If at any time the target current is exceeded, or the battery voltage drops by more than the specified amount, the test is stopped. Each sequence aims to run 10 tests up to the maximum, so even if the sequence is not completed, there should be a useful measurement amongst those taken.

Before each pulse, the battery voltage is measured. The DAC is set to provide the requisite current, and the pulse is applied by taking TESTCON high and waiting for 10ms. This gives time for the test conditions to stabilise. The current and voltage are measured, and TESTCON is taken low to end the test. The DAC voltage is also increased to its maximum to ensure that Q1 is switched off.

Scope 1 and Scope 2 show a sequence of eight test pulses. You can see the way the voltages change in the circuit as the test current is ramped up, from left to right. Scope 3 shows a single pulse and how the conditions settle before the voltages are sampled.

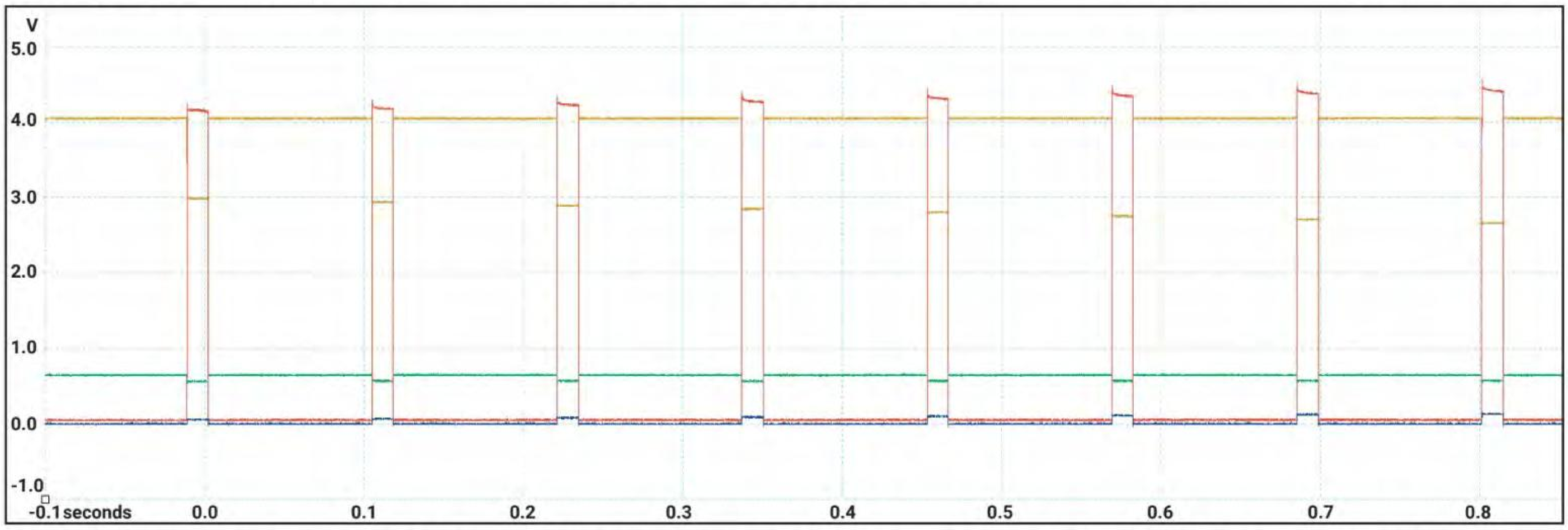
Features & Specifications

- Compact handheld unit
- Handles batteries/cells from 1V to 30V
- Test current up to 30A
- Battery connects via a pair of binding posts
- Reports test current, unloaded & loaded battery voltage, percent voltage drop & internal resistance
- Wiring & terminal resistance can be calibrated out
- Runs 10 tests up to a configurable maximum current
- Results appear on an OLED screen
- Self-protection built into the software
- Runs from a 9V battery or the battery being tested (above 7V)
- Operating current: 30mA
- Battery life: 10 hours plus with a standard 9V battery

The Checker displays the highest current reading that was made successfully, along with a measurement of the voltage drop and calculated internal resistance.

You can also view the results of the other samples taken (at lower current levels), as long as the Checker deems them valid.

It monitors for any conditions that might be problematic before each test.



Scope 2: the blue trace is VSHUNT, red is Q1's gate, green is Q4's base and yellow is the CURCON line, all per the scale on the left. This shows the increased drive to Q1's gate as the requested current increases and a fairly consistent voltage at Q4's base despite CURCON being driven at different voltages.

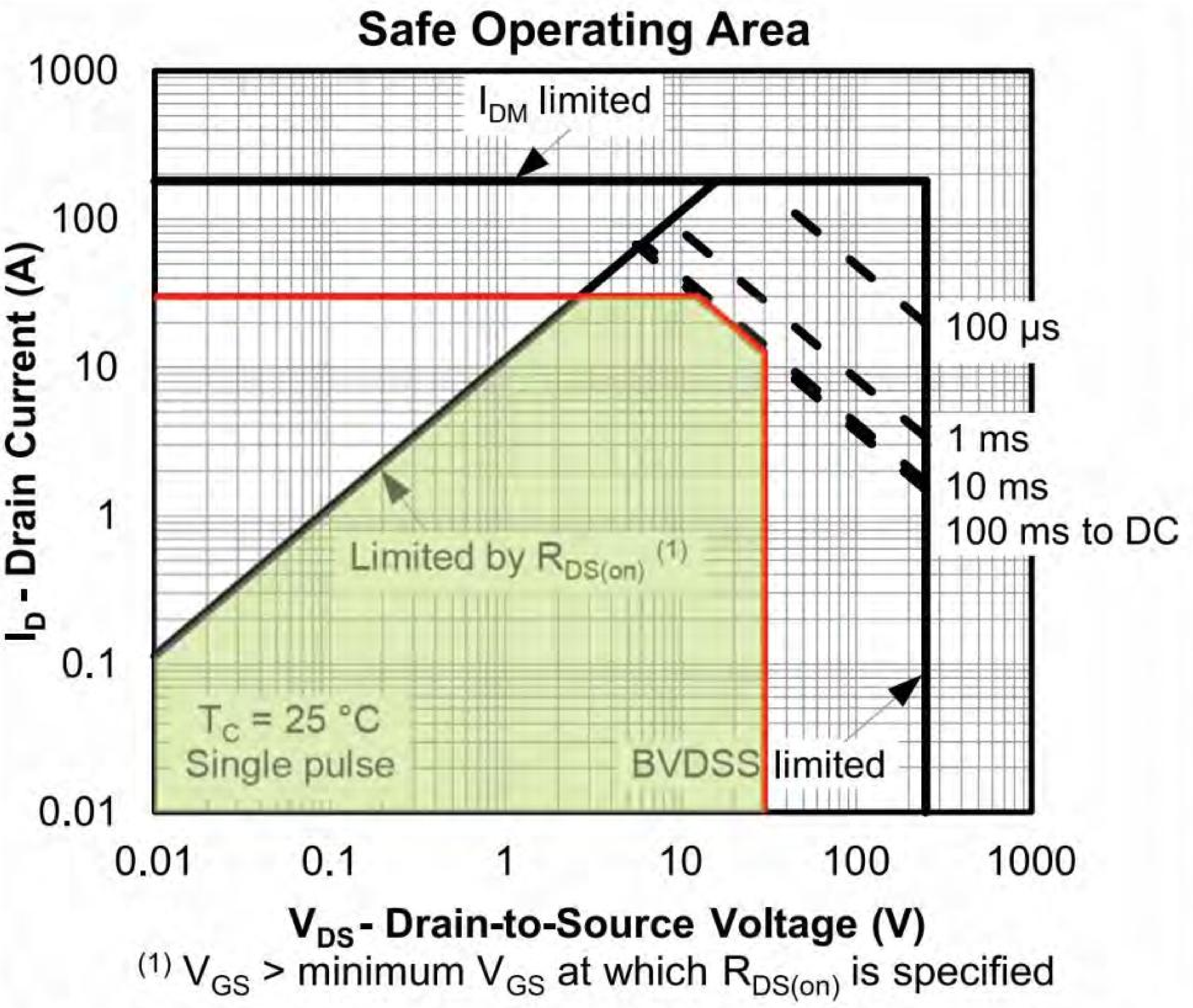


Fig.2: an annotated diagram from the SQM10250E data sheet; green shows the Mosfet's safe operating area, while the red line marks out the limits that are enforced by the software. The white area inside the red lines cannot be reached because of the Mosfet's minimum resistance.

For example, it checks that there is voltage on the BAT+ line before proceeding. There are also configuration pages to set some user preferences and calibration parameters.

The software also compares the maximum test current, and the measured battery voltage, against the Mosfet's safe operating area (SOA), shown in Fig.2. If the vertical line is exceeded, a "V too high" message is given. This can happen if the connected battery measures more than 30V.

For the diagonal line, which corresponds to a VI product of 400W, the Checker reports an "SOA error" and the calculated VI product value. It also calculates a lower test current that would be inside the safe operating area, based on the measured BUT voltage.

This means that for a fully charged 12V battery (at say 14.4V), the highest safe test current is around 27A, while a fully charged 24V battery can be tested up to about 13A. The horizontal line is enforced by not permitting the user to set the current over 30A.

Between tests pulses, a 100ms delay is inserted and after a test has been run, the software enforces a further delay proportional to the test VI product before allowing another test to begin. This ensures that there is negligible chance of the components overheating. Since it is when the results are displayed, you might not even notice it.

We'll examine some of the finer details of the software operation, including setup, calibration and usage after assembly is complete. The most critical of the Checker's calibration steps can be performed automatically, without any external test gear, and many of the others with just a multimeter.

Construction

The bulk of the construction involves populating the main PCB with surface-mounting parts, so the standard requirements for surface mounting gear apply. None of the parts are smaller than M3216/1206 or SOT-23, and the PCB is fairly spacious, so it is not too difficult to build.

You should ideally have a fine-tipped soldering iron (a medium tip should be OK), flux paste, a magnifier, tweezers and solder-wicking braid. Illumination and ventilation are also helpful. The power components sit on substantial copper areas, so you may need a reasonably powerful iron to solder them.

There are a handful of through-hole parts to add, then some cutting and drilling for the enclosure. The panel PCB is a bezel for the entire front of the enclosure, so not much precision is required when making holes in the enclosure.

Populating the PCB

The main PCB is coded 11104251 and measures 70 × 76mm. Follow along with the overlay diagrams (Figs.3 & 4) and photos. Pay attention to the transistors, since both NPN and PNP types are used in the same package. Care should also be taken that the diodes are fitted with the correct polarity. All the SMD parts mount on the same side of the PCB.

Start with IC1 and REG2, both in SOIC packages. If you have an MCP1804 (for REG1), then fit it instead of REG2. Apply flux to the pads on the PCB and rest the components in place. Both IC1 and REG2 have their pin 1 markers in the top left-hand corner.

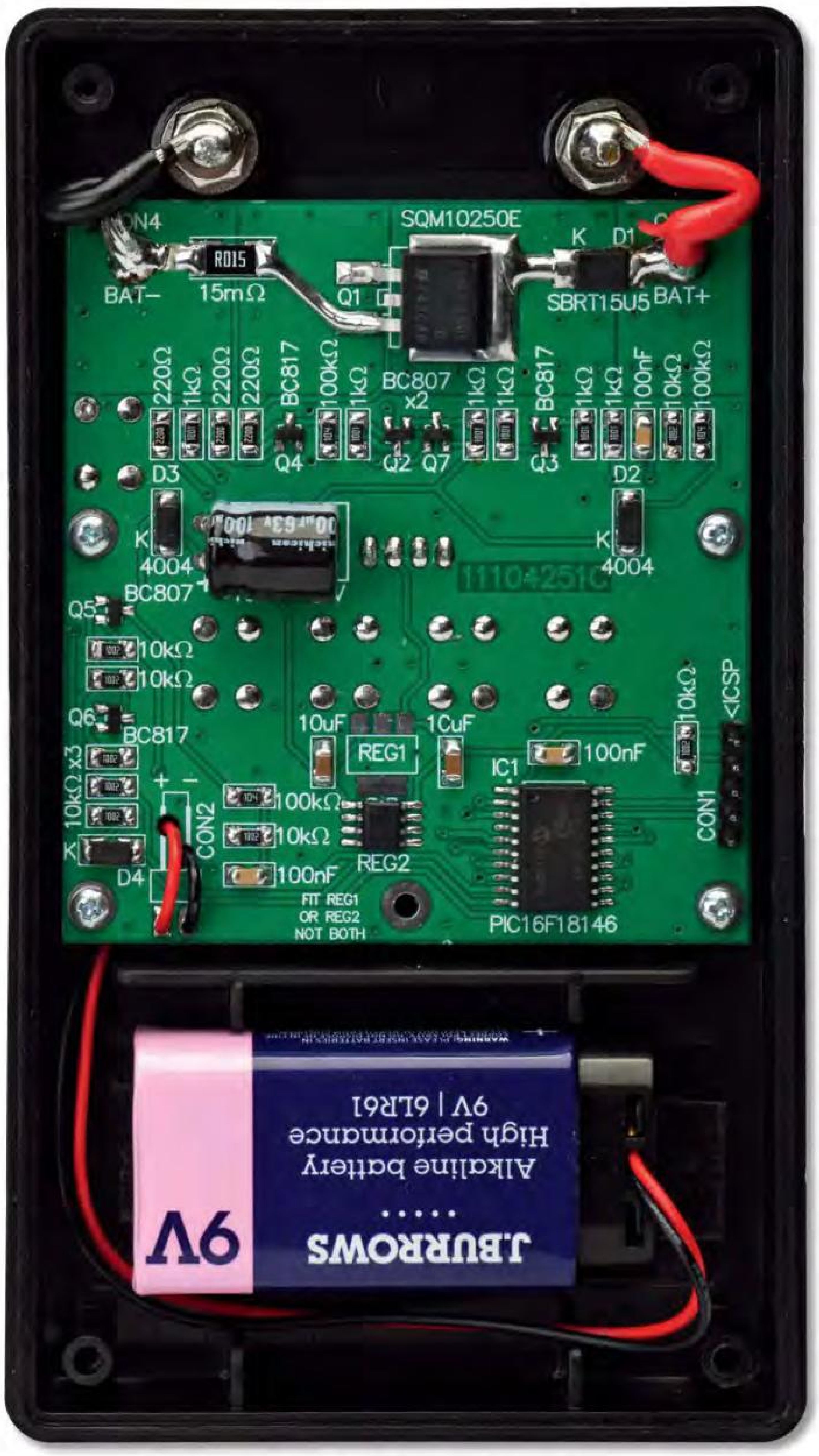
Clean your iron's tip and add a small amount of fresh solder. Tack one lead and adjust the parts with the tweezers until all the leads are located above their pads and the parts are flat against the PCB. Carefully solder the remaining pins, cleaning the iron and adding solder as needed.

If you bridge two pins, finish soldering the part before trying to fix the bridge. This will ensure it doesn't move out of position.

To remove a bridge, add a little flux paste, then press the braid against the bridge with your iron and allow it to draw up the excess solder. Go back and refresh any joins that you think might need it.

Next, add a thin layer of flux paste and solder the three BC807 PNP transistors: Q2, Q5 & Q7. These parts are smaller, but you can use much the same strategy as for the ICs. Follow with the three BC817 NPN transistors: Q3, Q4 & Q6. That will be all the parts in SOT-23 packages fitted.

Now fit the three 100nF capacitors, which should be thinner than the 10µF capacitors. They won't be marked except on their packaging. The two 10µF capacitors can be installed next, near the regulator. Now solder in the three smaller diodes, D2-D4. Pay attention to the cathode stripes and


CON4
SQM10250E
R015
BAT–
15mΩ
Q1
K
D1
SBRT15U5
BAT+
220Ω
1kΩ
BC817
100kΩ
BC807 x2
100nF
Q4
Q2
Q7
Q3
D3
D2
4004
BC807
Q5
10kΩ
Q6
10uF
1CuF
REG1
IC1
100nF
ICSP
CON1
10kΩ x3
CON2
D4
REG2
FIT REG1 OR REG2 NOT BOTH
PIC16F18146
11104251C
9V
High performance
Alkaline battery
J.BURROWS
9V | 6LR61

The back of the Checker just before the rear panel is screwed on. The binding posts connect to the main PCB with short lengths of heavy-duty insulated wire, and the main current carrying path is supplemented with extra solder.

ensure they are placed closest to the ‘K’ markings on the PCB silkscreen.

The 20 M3216/1206 size (3.2 × 1.6mm) resistors are the last of the smaller parts. Check the value of each using a multimeter (set to resistance) or by visually examining the resistance code and making sure it matches the value printed on the PCB silkscreen or overlay diagram.

If your iron has adjustable temperature, you can now turn it up for soldering the three larger parts: D1, Q1 and the 15mΩ shunt. The strategy is much the same, although you might need to apply more heat, which could take more time.

For Q1, the gate pin (at top left, above the Q1 silkscreen marking) will have less attached copper, so we recommend you tack it first. Also make sure you spread flux paste on all the pads before placing the part, so that solder will flow under it later.

Q1 will only fit one way, but you should check the polarity of D1. D1’s pads and leads are also asymmetrical, so you can match the two smaller leads to the smaller pad on the PCB. The shunt resistor is not polarised.

Once you have the components secured on all leads, you can check that there is no continuity between the BAT+ and BAT– pads in either direction. If this is low resistance or it shows a low voltage on a diode test, you may have the diode reversed. You should measure around 100kΩ due to the sensing divider.

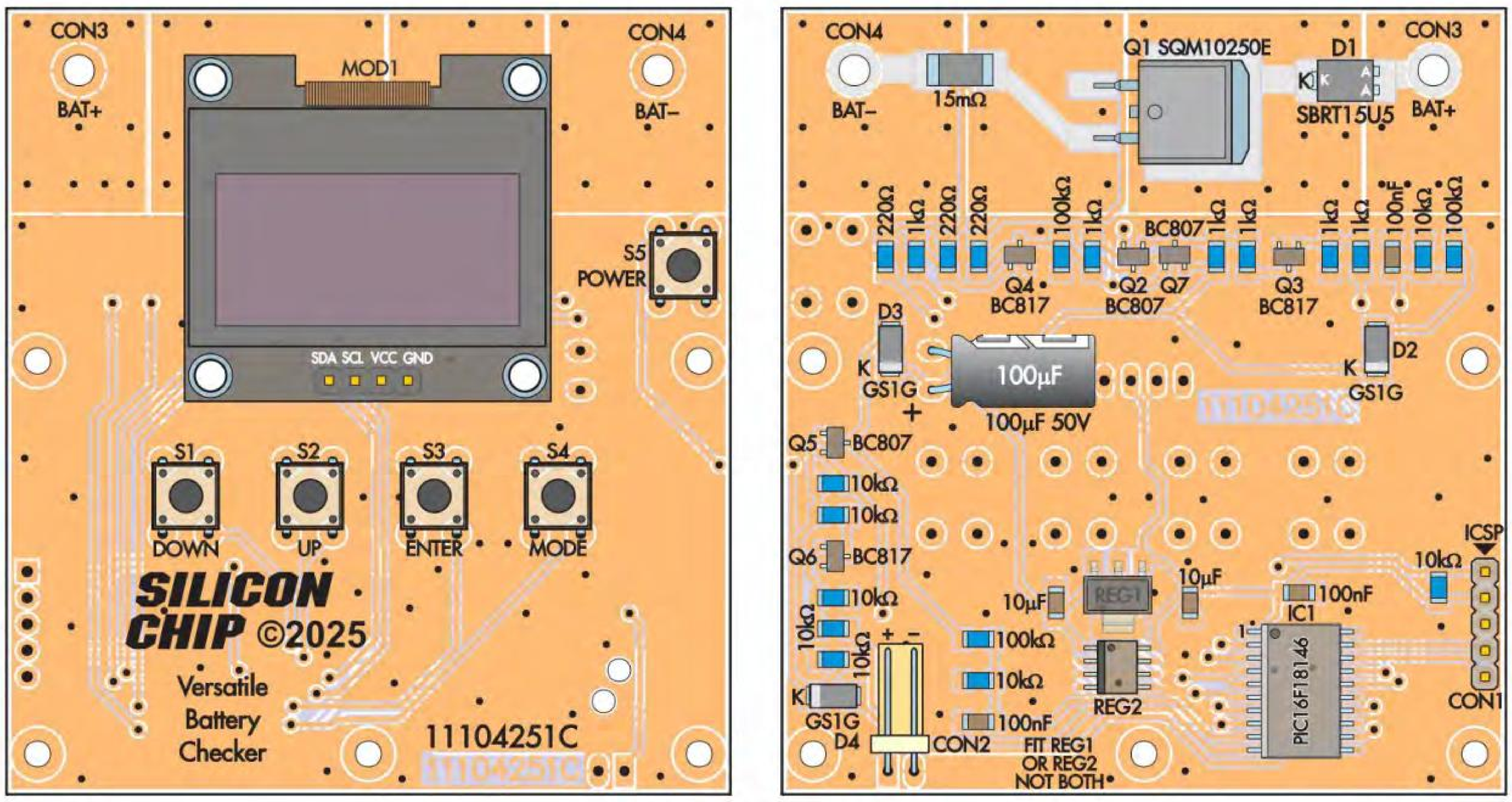
If all is well, you can add more solder to the exposed copper areas, which will enhance their current-carrying capacity (shown in grey near the top of Fig.4). Then use a flux solvent or isopropyl alcohol to clean up the PCB and allow it to dry fully. Check the PCB thoroughly for solder bridges and other defects and repair as needed.

Programming IC1

At this stage, there is enough circuitry attached to IC1 that it can be programmed if that is required. If you have purchased a pre-programmed microcontroller from the Silicon Chip Shop, this will not be necessary.

CON1 must be fitted to allow a programmer to connect. It goes on the same side as the SMD components, and can be left in after programming, since it won’t foul the case. You can see it in our photos, since we used CON1 quite a lot during development.

You’ll need the Microchip IPE (integrated programming interface) software. This is a free download as part of the MPLAB X IDE from the Microchip website at www.microchip.com/en-us/tools-resources/develop/mplab-x-ide


CON3
BAT+
MOD1
CON4
BAT–
S5
POWER
SDA SCL VCC GND
S1
S2
S3
S4
DOWN
UP
ENTER
MODE
SILICON CHIP ©2025
Versatile Battery Checker
11104251C
CON4
Q1 SQM10250E
D1
CON3
BAT–
15mΩ
K
SBRT15U5
BAT+
220Ω
1kΩ
100kΩ
BC807
Q4
BC817
Q2
Q7
Q3
100nF
10kΩ
D3
D2
K
GS1G
100µF
100µF 50V
Q5
10kΩ
Q6
REG1
10µF
ICSP
100nF
IC1
100kΩ
REG2
CON1
D4
CON2
FIT REG1 OR REG2 NOT BOTH
PIC16F18146

Figs.3 & 4: all the SMD parts are on one side of the PCB and should be installed first. Only one of REG1 or REG2 should be fitted. The exposed traces in the high-current section of the circuit near the top can be supplemented with extra solder. The tactile switches are fitted to the other side of the PCB, as is the OLED screen.

You'll also need a programmer like a Snap, PICkit 4 or PICkit 5. If your programmer cannot supply power to the circuit, then the easiest way will be to rig up something to supply 6V or more to CON3 and CON4 (observing their respective polarity markings). A current-limited supply set to 50mA is ideal, since the circuit should not draw more than that when operating.

Connect the programmer and, in the IPE software, select the PIC16F18146, open the HEX file and press the Program button. Check that the programming completes and the file is verified successfully.

Nothing will happen after that, since there is no display connected. Disconnect the programmer and power supply before proceeding.

Case cutting

You'll need to cut the holes to allow the OLED (MOD1) to be correctly located relative to the front panel. The front panel PCB can be used as a template for the holes. The seven round holes should match the front panel closely, while the square hole for the OLED will need to be cut larger on the case to allow the display to sit the directly behind the panel.

Parts List – Versatile Battery Checker

1 double-sided PCB coded 11104251 measuring 70 × 76mm
1 double-sided 0.8mm-thick black PCB coded 11104252 measuring 131 × 68mm (front panel)
1 Retex Betabox 33050552 145 × 80 × 34mm handheld enclosure with battery compartment
5 through-hole SPST tactile switches with stems 9mm above the PCB (S1-S5, 6mm actuator length)
1 1.3in I^2C OLED module (MOD1) **[Silicon Chip SC6511 or SC5026]**
1 9V battery and battery snap
1 5-pin header, 2.54mm pitch (CON1; **optional**, for ICSP)
1 2-way right-angle 2.54mm polarised header and matching plug (CON2; **optional**)
1 red binding post (CON3)
1 black binding post (CON4)
4 self-adhesive small rubber feet
1 piece of double-sided tape to secure battery
1 fresh AA cell and holder for setup and testing
1 5cm length of red 25A+ rated wire
1 5cm length of black 25A+ rated wire
1 small tube of neutral cure silicone or similar resilient glue

Semiconductors
1 PIC16F18146-I/SO 8-bit microcontroller programmed with 1110425A.HEX, SOIC-20 (IC1)
1 MCP1804-5 5V low-dropout linear regulator, SOT-223 (REG1) ***OR***
1 TLE4269G 5V low-dropout linear regulator, SOIC-8 (REG2)
1 SQM10250E 250V 65A N-channel automotive-grade Mosfet, D2PAK-3 (Q1)
3 BC807 50V 800mA PNP transistor, SOT-23 (Q2, Q5, Q7)
3 BC817 50V 800mA NPN transistor, SOT-23 (Q3, Q4, Q6)
1 SBRT15U50SP5 50V 15A schottky diode, POWERDI-5 (D1)
3 M4/GS1G/SM4004 400V 1A diodes, DO-214AC (D2-D4)

Capacitors
1 100µF 50V radial electrolytic
2 10µF 50V SMD M3216/1206 size X5R ceramic
3 100nF 50V SMD M3216/1206 size X7R ceramic

Resistors *(all M3216/1206 size 1% ⅛W unless noted)*
3 100kΩ
8 10kΩ
6 1kΩ
3 220Ω
1 15mΩ M6331/2512 size 1% 3W

Versatile Battery Checker Kit (SC7465, ~£40 + postage):
Includes everything in the parts list (and the case) except the optional components, batteries and glue.

Fig.5 shows the required cut-outs. This is shown from outside the case, as you will only be able to mark the case from the outside using the panel. It won't matter too much if you the mark the case since the panel will cover it.

None of the holes need to be cut with any accuracy since the panel PCB will hide any imperfections. Still, it is not hard to cut the round holes accurately, and they can be enlarged if needed. You can see the general layout in our photos.

If things don't quite line up as you are fitting the through-hole parts in the next step, you can adjust the case as long as the panel hasn't been glued to it yet.

Through-hole parts

Solder the five tactile switches now, noting that they are on the opposite side of the PCB to the surface-mounting parts. Ensure that they are flat against the PCB so that they point straight up through the holes in the front panel.

Fitting MOD1 requires a bit of precision, since it needs to be placed just behind the panel PCB for the best result. To align it, screw the main PCB into the case. The tactile switches should neatly pass through their respective holes without binding.

If the header has not been soldered to the OLED module, do that now, keeping it as square as possible and ensuring it does not protrude above the front of the screen.

Alternatively, if the header is already fitted, you might find that the pins protrude slightly above the screen. In that case, you can trim them back with some nippers.

Slot the OLED module into place but do not solder it yet. Tape the front panel PCB temporarily in its location to allow the OLED module to be positioned correctly. We want to have the OLED sit just behind the panel and flat against it.

You should be able to rest the assembly flat on its face and allow the OLED to rest against the back of the panel PCB. Tack one pin with your iron and check that it looks aligned from the front. It should be parallel to and just behind the panel. You can also check that it is square by com-

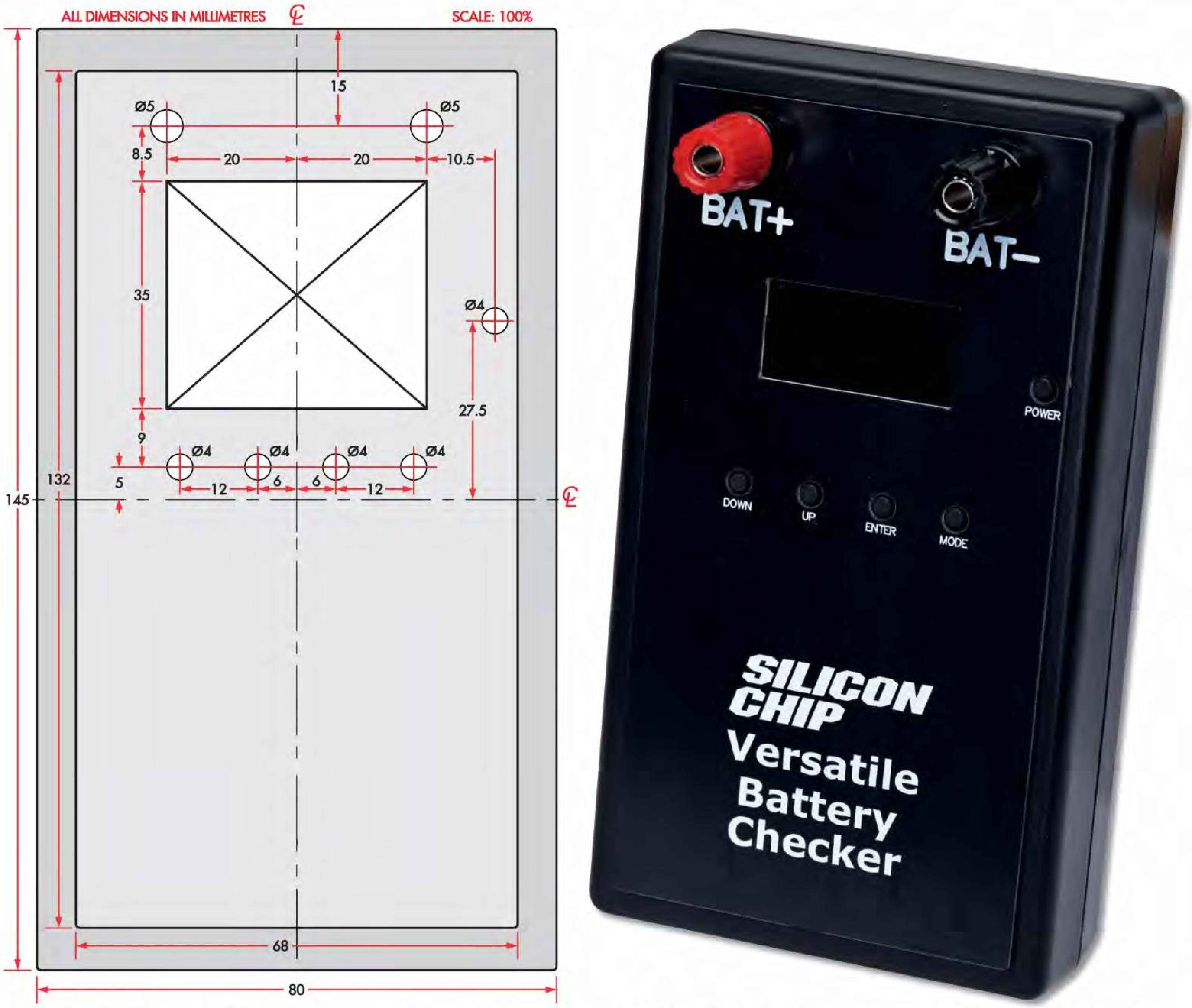


Fig.5: the front panel PCB can be used as a template for the round holes. The OLED screen is smaller than the rectangular cutout, but it's needed to allow the OLED to sit just behind the front panel. The Versatile Battery Checker is a handy tool for checking the condition of all manner of batteries. It can deliver test pulses up to 30A and handle batteries with up to 30V at the terminals. Internal resistance and percentage voltage drop are shown at the conclusion of each test.

paring the OLED's outline against its silkscreen markings on the main PCB.

If all is well, solder the remaining leads and detach the PCB from the case. The 100µF capacitor can be fitted now. Bend its leads 90°, paying attention to the polarity markings, solder it to the PCB and trim the leads, keeping the offcuts.

There are also two larger pads on the main PCB under the OLED. Thread the offcuts through the holes in the OLED module and solder the lower end to the large pad on the PCB. Then solder the tops of the offcuts to the OLED and trim them to a tidy length.

If you are using the plug-and-socket arrangement for CON2 (the 9V battery), these can be installed now too, with the battery snap wires crimped into the socket. Otherwise, thread the wires for the battery snap through the holes in the PCB (to give a degree of strain relief) and then solder them to their respective pads, observing the polarity. That is how we built the prototype.

Binding posts

You can perform a basic functional test of the Checker by connecting a 9V battery now. Nothing should happen until you press S5, the power button. The OLED should illuminate and show something like Screen 1 and then Screen 2. The UP and DOWN buttons should change the test current value on the third line. Disconnect the battery before continuing with assembly.

Reattach the PCB to the case, then rest the front panel in place. It should locate itself fairly accurately within the boss around the edge of the enclosure. Spread a thin film of neutral-cure silicone sealant around the lower half of the case and secure the top half of the panel with the binding posts.

Make sure to fit the red binding post near CON3 and the black binding post near CON4. Solder the red and black wires to their respective pads on the PCB, then clamp the lower half of the case to the panel until the silicone has cured.

You can also apply some silicone to the 100µF capacitor so its leads don't flex too much. Fit the 9V battery (if using it), affix it with the double-sided tape and screw the back onto the enclosure.

Testing and calibration

The operation of the Checker is shown in Screens 1-16. You can see that there are some parameters that can be calibrated, but only a few are absolutely necessary. A 1.5V cell such as a fresh alkaline AA type is used as our calibration BUT (a low-voltage battery with limited current capacity is less likely to cause damage if there is a problem).

Connect the AA cell to CON3 and CON4 with the correct polarity. Press S5 (POWER) to switch on the Checker. The same button switches it off, although it might not respond right away if it is in the middle of a test or other operation. Hold S5 until "OFF" is shown, then release it.

You should see something like Screen 2, but with the second line showing around 1.5V. The top line should be close to 9V. If these values are markedly different, there might be a problem. In that case, power off the Checker and examine the PCB for assembly errors.

To calibrate the Checker, hold MODE for a second until the screen blanks, then release MODE. You will see Screen 6. With the AA cell attached to CON3 and CON4, press ENTER to run the calibration.

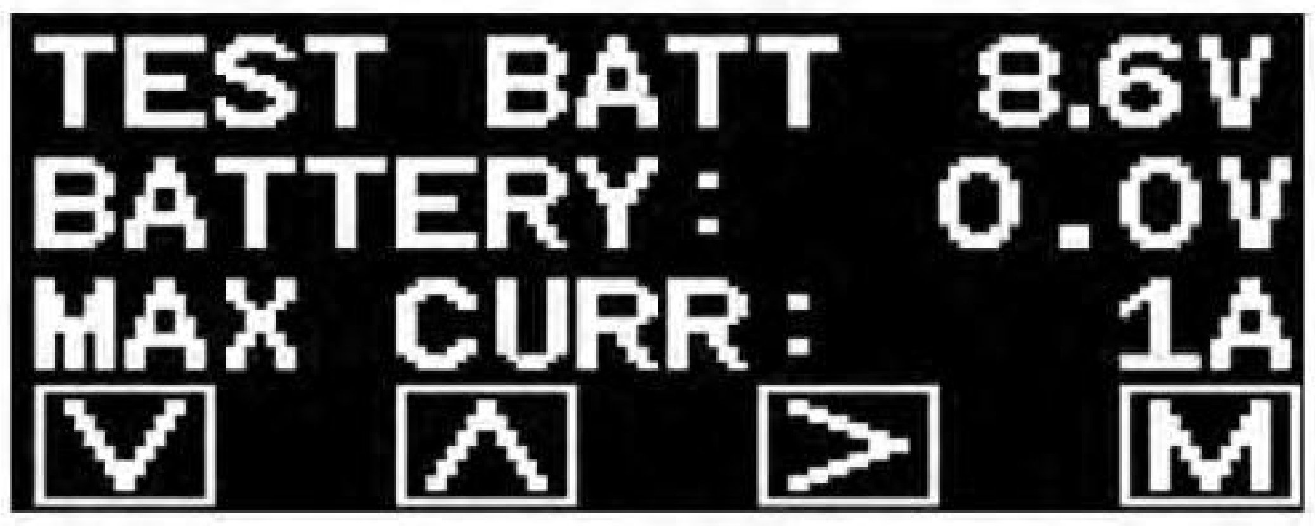
This scans through the DAC settings to find the lowest value that will activate Q1 and sink 1A. You can also trim this manually with the UP and DOWN buttons.

If you see a "Scan Failed Check Battery" message, make sure you have a fresh cell. It should be able to deliver 1A without dropping by more than 10%; we wouldn't trust any modern AA cell that struggles with this! Any other battery that the Checker can test should work for the purposes of this calibration.

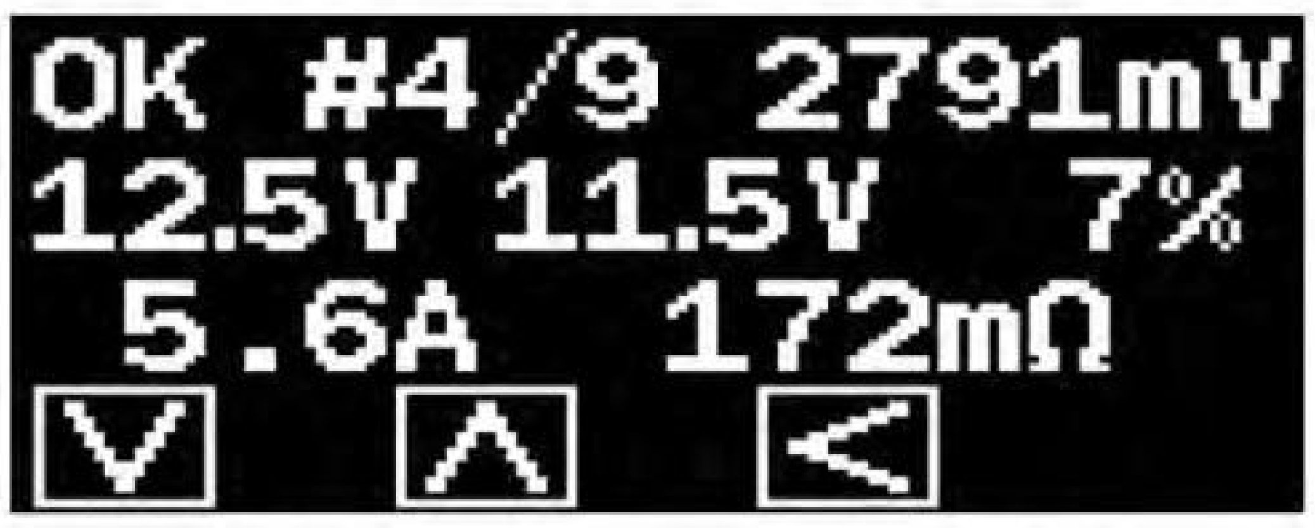
Press MODE repeatedly until you return to Screen 2 and run a test at 1A by pressing ENTER. After about a second, you should see Screen 4. Scroll through the test results with the up and down buttons; there may only be one or a few. You should see a result showing a current around 1A or lower if the voltage has dropped over 10% at a lower current level.

SILICON CHIP
VERSATILE
BATTERY CHECKER

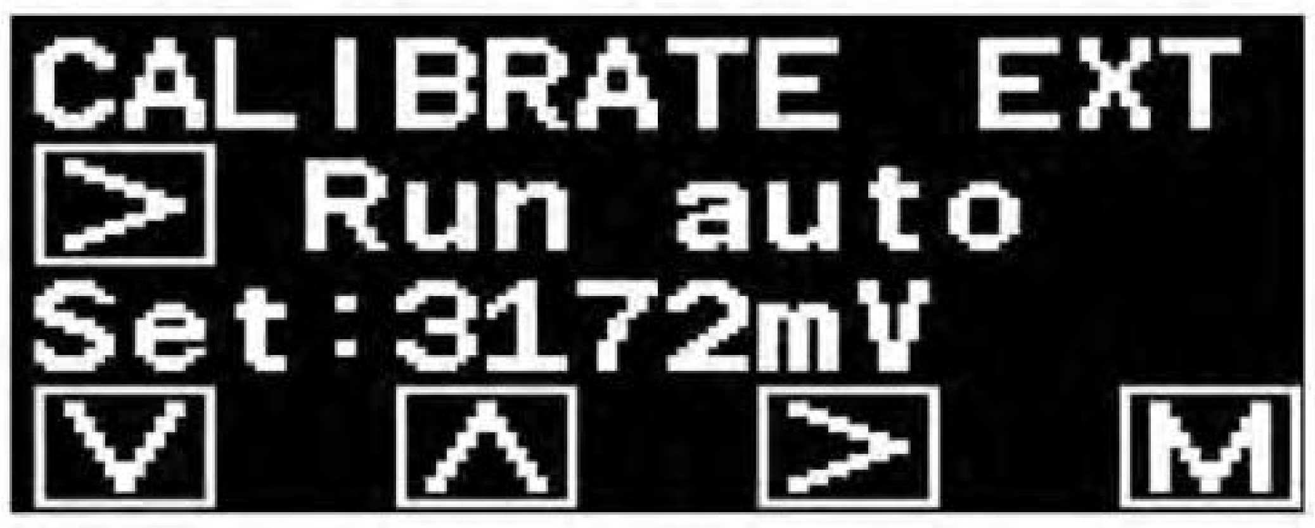
Screen 1: you should see this splash screen when the Checker is powered on before it switches to the main operating screen (Screen 2 or 3).


TEST BATT 8.6V
BATTERY: 0.0V
MAX CURR: 1A

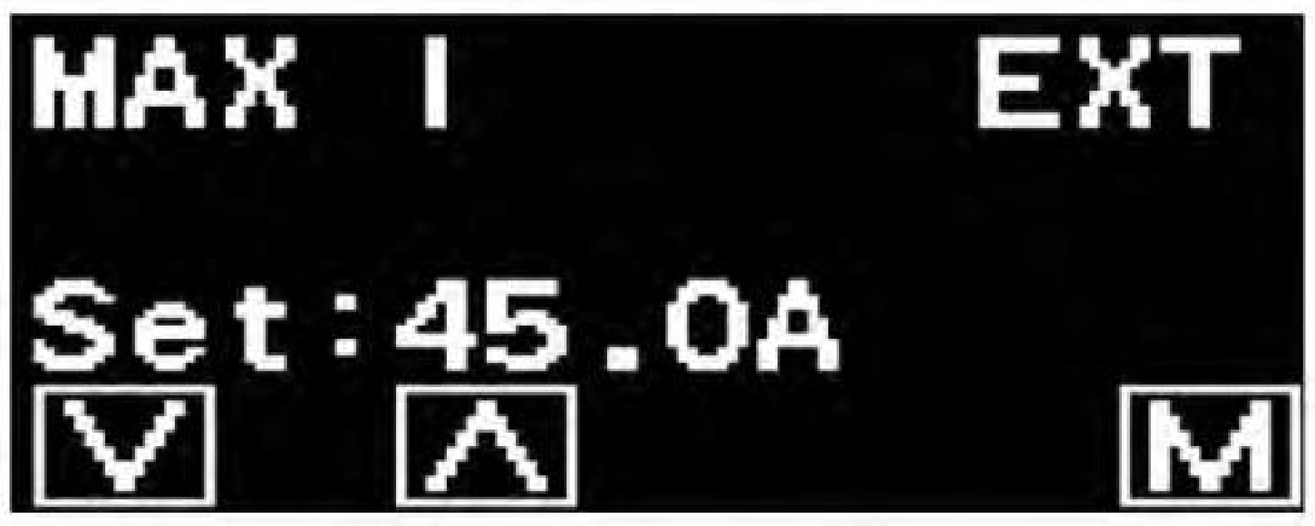
Screen 2: when running from the internal 9V battery, its voltage is shown at upper right. The down and up icons indicate that S1 and S2 can be used to adjust the test current.


OK #4/9 2791mV
12.5V 11.5V 7%
5.6A 172mΩ

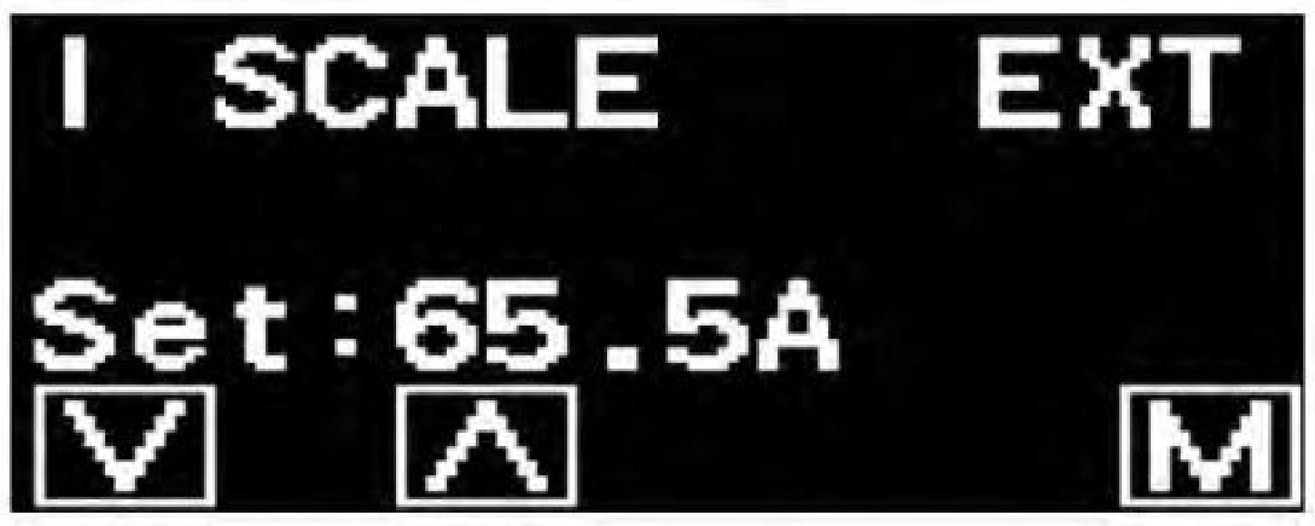
Screen 5: the no-load and loaded voltages, along with the calculated percentage drop, are shown on the second line. Below are the test current and calculated internal resistance.


CALIBRATE EXT
> Run auto
Set:3172mV

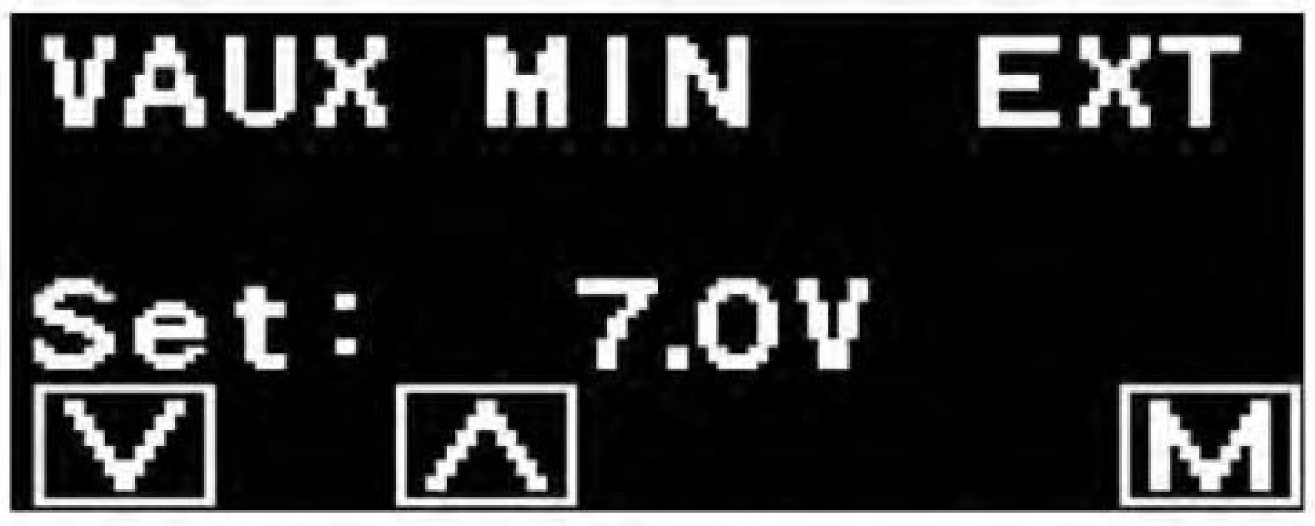
Screen 6: holding S4 for a second opens the setup menu. The first page shows the zero-current DAC setting level. Briefly pressing S4 again cycles through the remaining menu items.


MAX I EXT
Set:45.0A

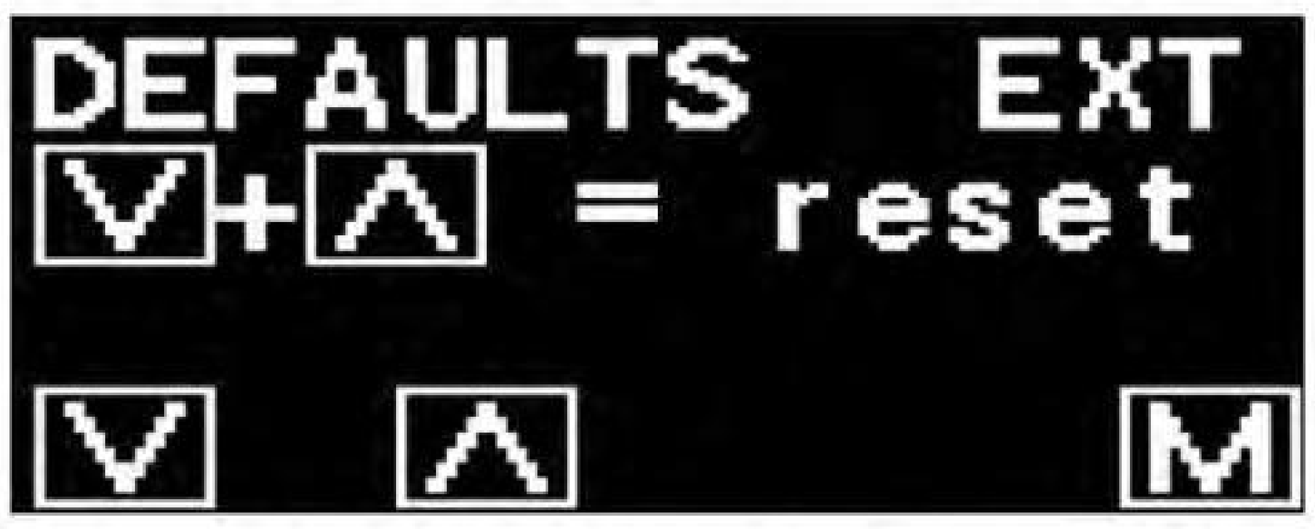
Screen 9: this value sets the nominal target current when the DAC is set to 0V. It is used to calibrate the target current setting during tests.


I SCALE EXT
Set:65.5A

Screen 10: this calibration sets the scaling for current measurements. All calibration and configuration values are saved to EEPROM and used immediately.


VAUX MIN EXT
Set: 7.0V

Screen 13: this is the lowest voltage that will allow the Checker to operate tests; below this, the circuitry cannot guarantee that the Mosfet will be driven hard enough.


DEFAULTS EXT
V+Λ = reset

Screen 14: if there are problems with the calibration and configuration values in EEPROM, they can be reset by pressing S1 and S2 simultaneously on this screen.

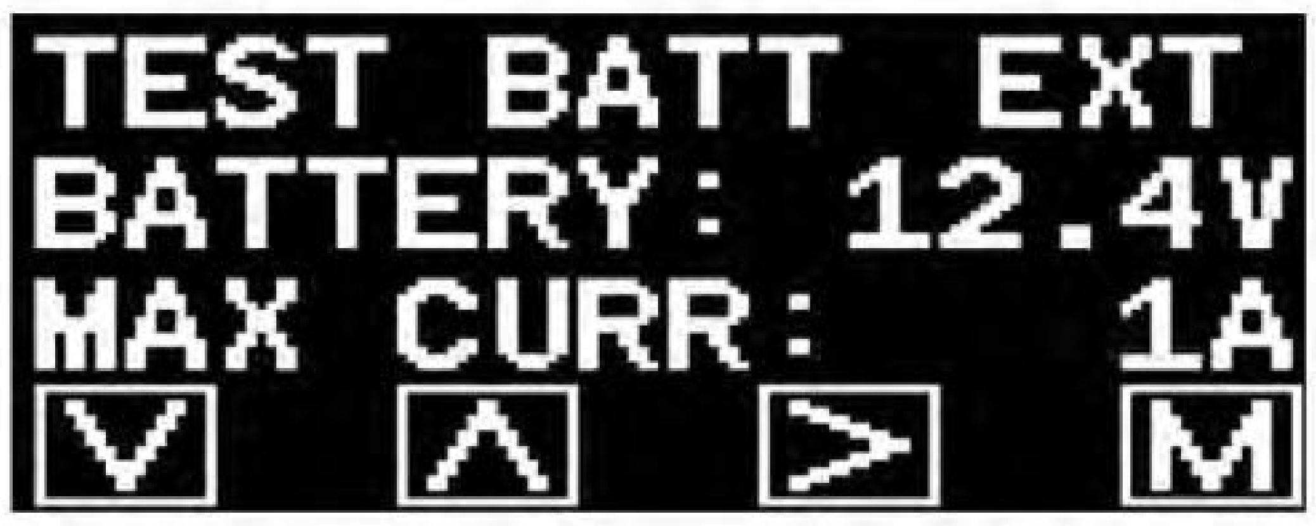
This indicates that the Checker is basically functional. You can try the Checker on other batteries if you like, to test the maximum current setting. If possible, run some tests at a higher current like 20A. One AA cell probably can not do this! Perhaps you could use a car battery, or a pack from a remote-controlled vehicle.

A current-limited power supply can also be used to run the Checker through its paces.

You'll see that the top line shows the test number (0-9). If the maximum current is well calibrated, and the BUT can supply the test current without sagging more than 10%, then test #9/9 should be very close to the target test current.

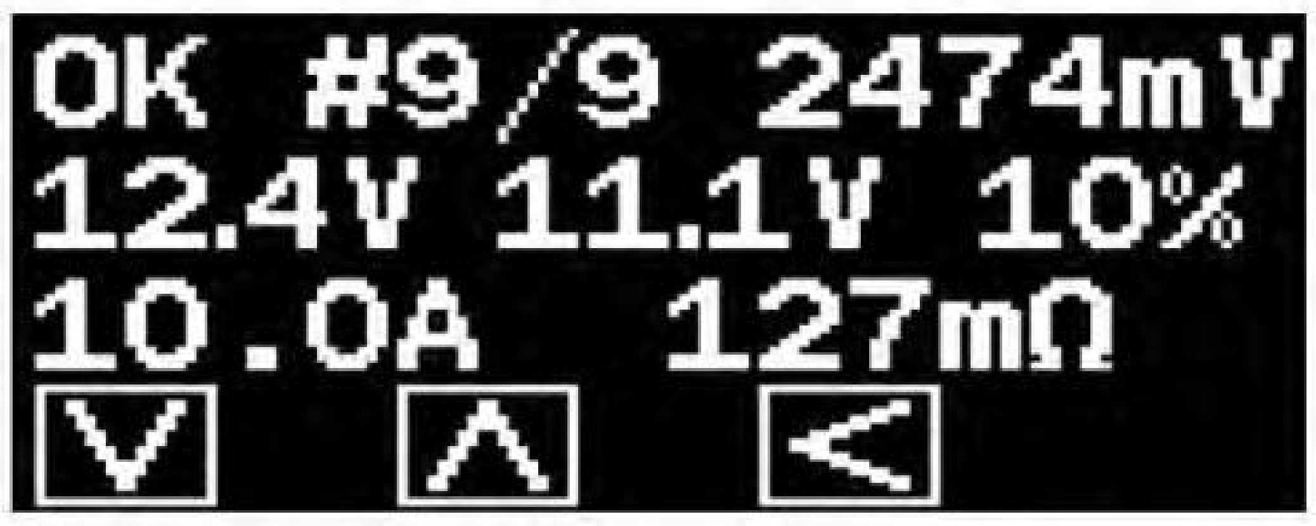
If not, you can trim the MAX I parameter to adjust this. Reduce it if the measured current is too high and increase if it is too low. Do this in small steps and run a few tests after each adjustment to get a feel for how much the results will vary.

Even after making an adjustment, the Checker may overshoot the maximum current slightly, by less than an amp. This is due to the limited DAC resolution. If you prefer to avoid this, you



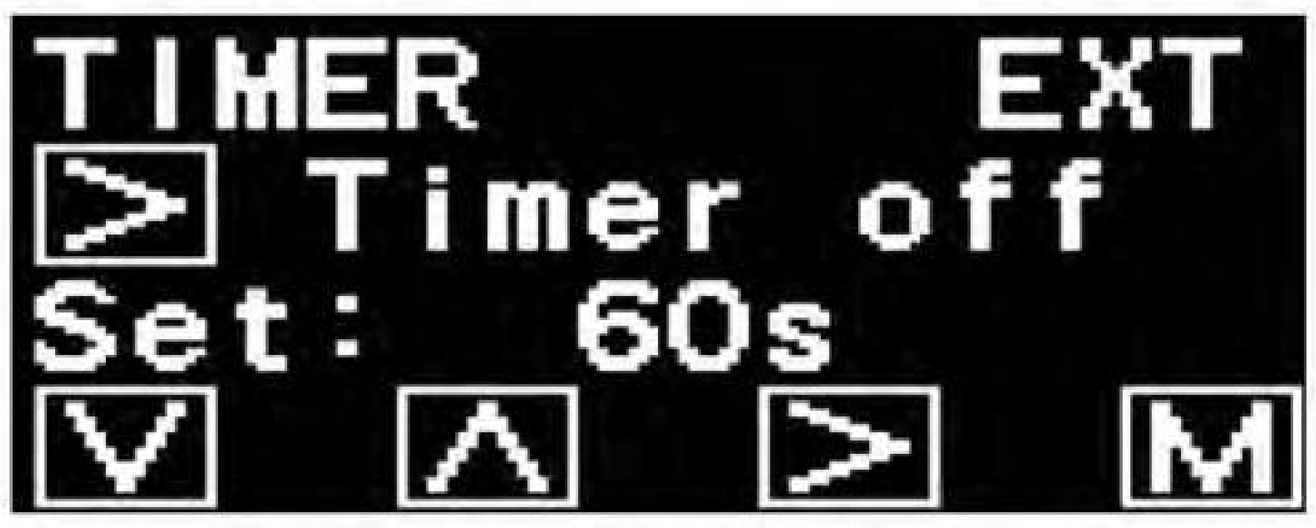
```
TEST BATT  EXT
BATTERY: 12.4V
MAX CURR:   1A
[V]  [Λ]  [>]  [M]
```

Screen 3: when powered from a battery connected to CON3/CON4, the Checker shows "EXT" at top right. The right arrow icon above S3 starts a test cycle.



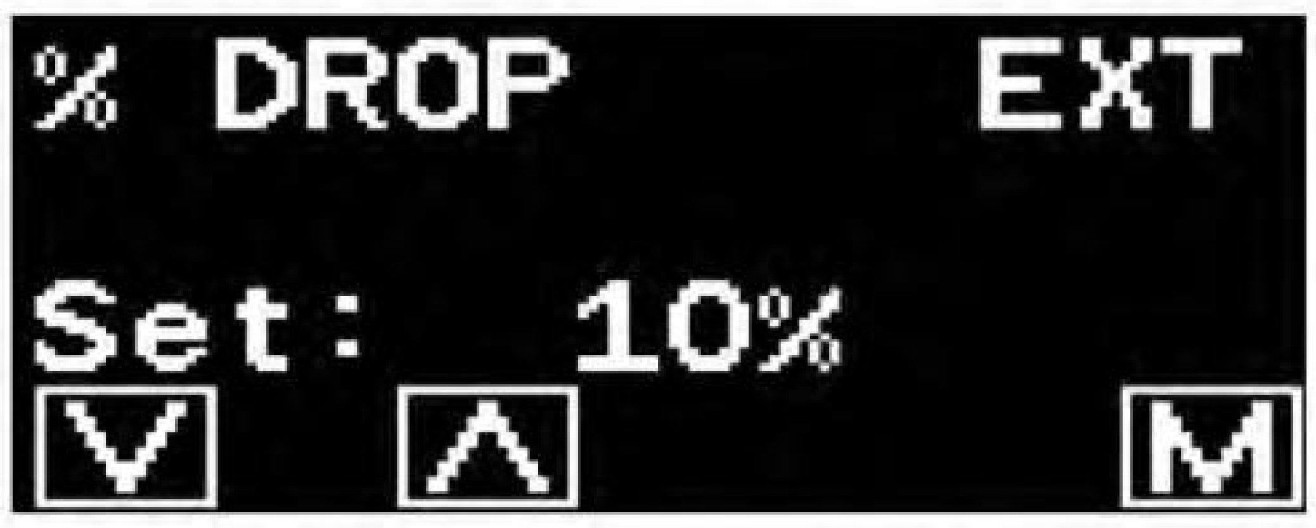
```
OK #9/9 2474mV
12.4V 11.1V 10%
10.0A  127mΩ
[V]  [Λ]  [<]
```

Screen 4: the test results are shown after about a second, with the first line showing the number of successful tests. UP and DOWN can be used to cycle through the other test results.



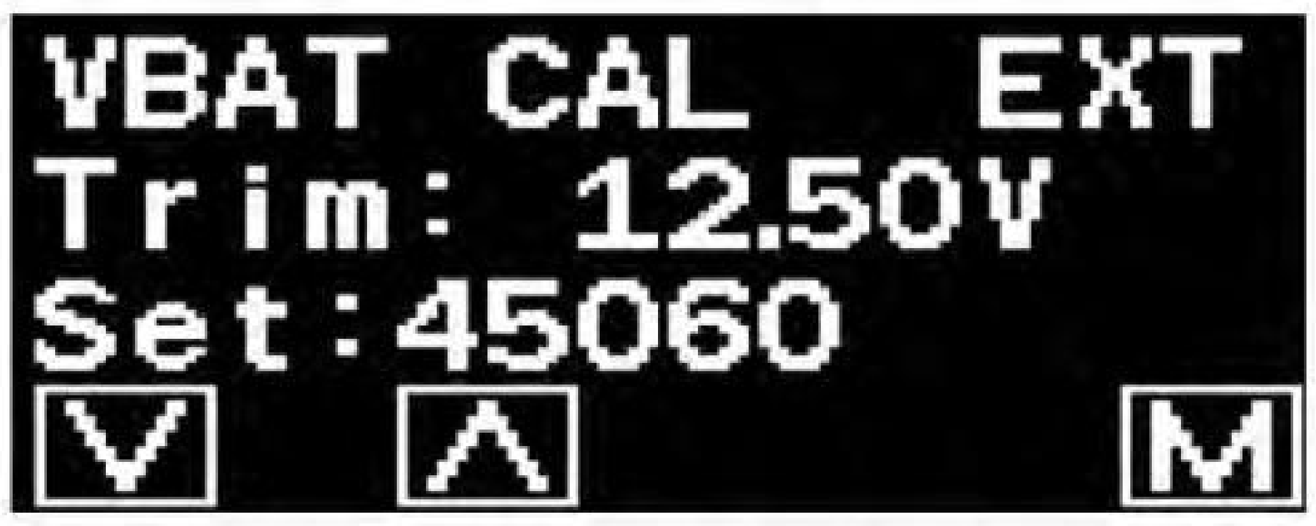
```
TIMER       EXT
[>] Timer off
Set:  60s
[V]  [Λ]  [>]  [M]
```

Screen 7: the timer for internal battery operation is set here with S1 and S2, and enabled or disabled with S3. If the timer is disabled, then the Checker will not automatically power off.



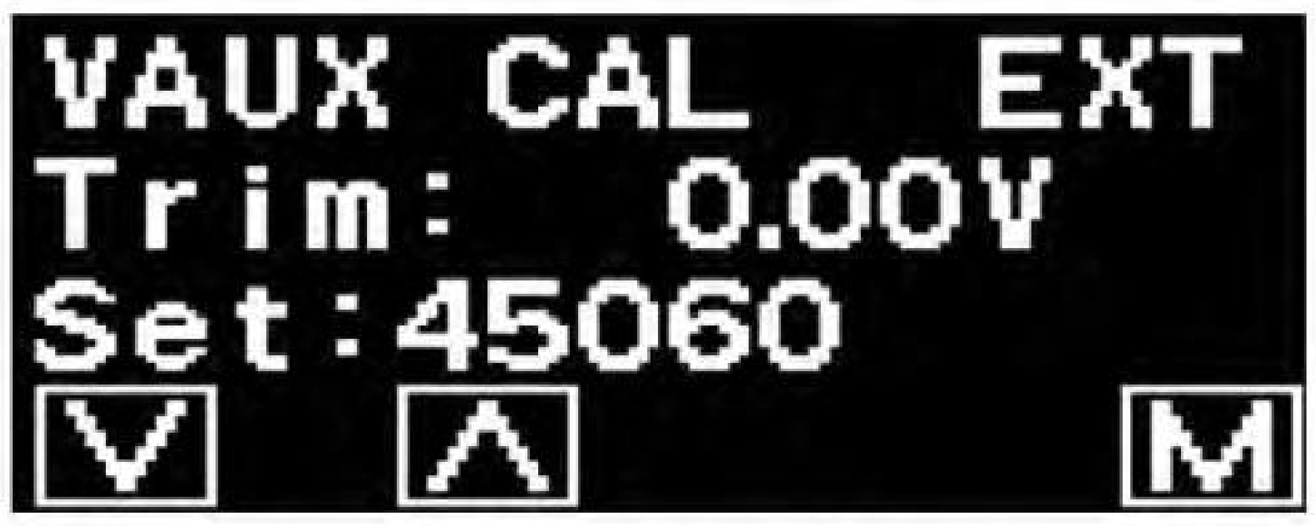
```
% DROP      EXT

Set:  10%
[V]  [Λ]       [M]
```

Screen 8: the maximum voltage drop is set here, in percent. If the Checker detects a drop higher than this, it will stop the test, even if it hasn't reached the maximum test current.



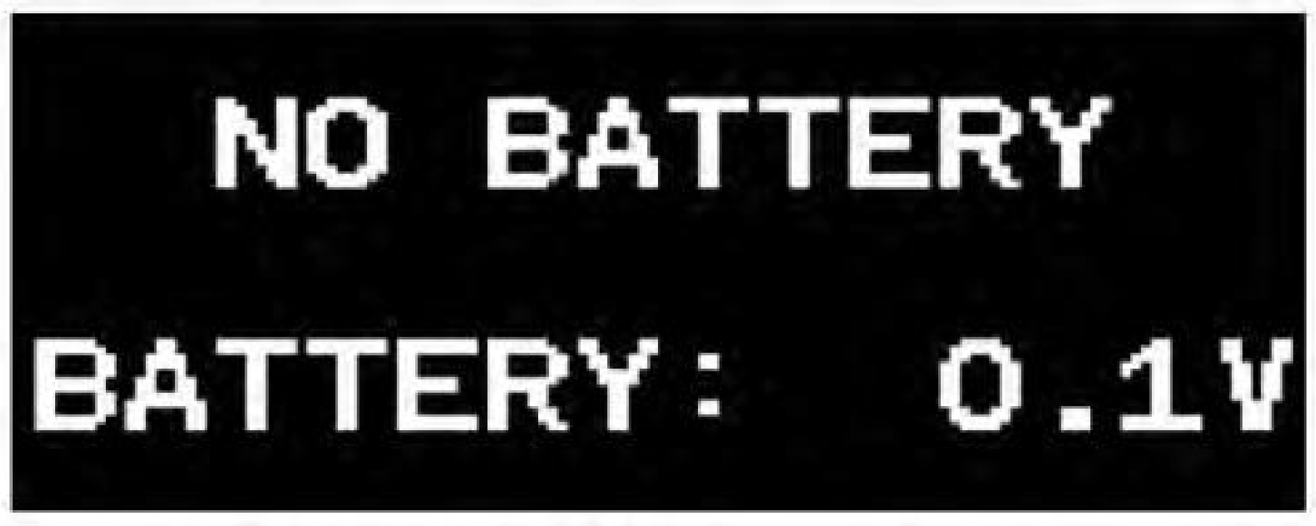
```
VBAT CAL   EXT
Trim: 12.50V
Set:45060
[V]  [Λ]       [M]
```

Screen 11: use a multimeter to trim the calibration factor here so that the displayed value matches the voltage of the battery attached to CON3 and CON4.



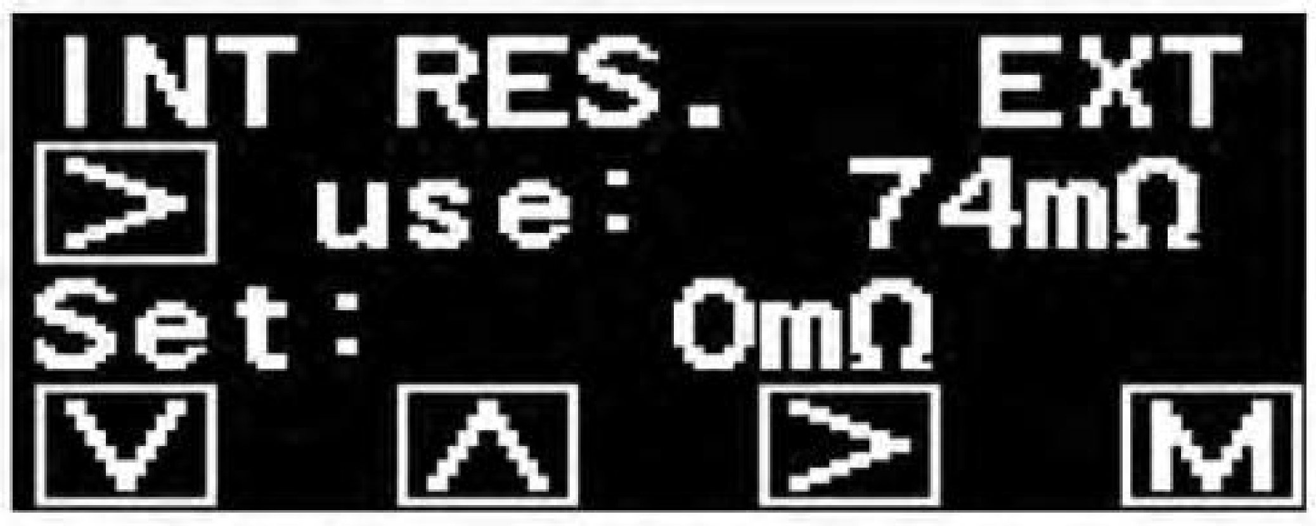
```
VAUX CAL   EXT
Trim:   0.00V
Set:45060
[V]  [Λ]       [M]
```

Screen 12: the VAUX calibration works much the same as the external battery calibration seen in Screen 11. If in doubt, you can use the same calibration factor.



```
  NO BATTERY

BATTERY:  0.1V
```

Screen 15: one of the error messages that might be seen when there is a problem. This will appear if you try to run a test without a battery connected to CON3 and CON4.



```
INT RES.   EXT
[>] use:   74mΩ
Set:       0mΩ
[V]  [Λ]  [>]  [M]
```

Screen 16: the offset applied for the intrinsic resistance of the Checker and its wiring is set here. You can either use the value from the latest test or adjust it up and down manually.

can set the MAX I value even lower to be more conservative.

Error messages

You might see a few error messages when running tests. These are generally intuitive, although, for example, a "V too low" message can sometimes be fixed by trying a lower test current. This message means that even the lowest test current of the test set caused an excessive voltage drop. You could also check the calibration.

If the 9V battery is getting flat, you will see the voltage dropping on the initial screen. When it drops below 7V, you might see a message reading "AUX/BAT low". That means it is definitely time to fit a new 9V battery. The "SOA ERROR" message should go away if you reduce the test current and try again.

"I too high" probably means that the calibration is off and the Checker could not reach the target current using the settings it has. There is also an option to reload the default configuration from flash memory if they do somehow end up corrupted or unusable. You might see "SETTINGS ERROR" if the Checker thinks there is a problem with the configuration.

Usage

The captions for the Screens give more details about the other configuration and calibration settings, but it will work quite well without much setup. Run a few tests with the Checker to try out its operation and you should become familiar with how it works.

From the initial page, dial in the maximum desired test current and press ENTER to start the test. Wait for the results and use the UP and DOWN buttons to scroll through them, then press ENTER to return to the initial page.

If you will only be using the Checker with 12V batteries or higher, the 9V battery can be left disconnected. The Checker will power up from CON3/CON4 if it can, so you can simply hook it up to a BUT, run a test and then disconnect the battery. The target test current is saved in EEPROM and reloaded when the Checker starts up.

We found normal internal resistance values fairly easy to find for reputable brands of batteries. For example, an alkaline AA cell should measure around 150mΩ. An 18650 lithium cell should be under 100mΩ. A 7Ah SLA battery like Jaycar's SB2486 is specified at 25mΩ, while a car starting battery should be even lower (under 10mΩ).

Naturally, if a battery or cell reads much higher than specified, it should be considered for replacement.

Nulling the wiring resistance

Some 12V lead-acid battery chargers estimate battery internal resistance (in mΩ) using the equation 3000/CCA (cold cranking amps). A reasonably large car battery will typically be rated at 600CCA, implying a 5mΩ internal resistance.

The Checker has a calibration value for the intrinsic resistance of the Checker and its wiring; this is an offset in milliohms that is subtracted from all calculated internal resistance values. The default value is 0mΩ, so measurements will display unadjusted values unless you change this.

Screen 16 shows how this can be edited. It can be manually trimmed up and down, or it can use the value of the most recent test that has occurred. Thus, a simple way to calibrate this value is to run a test on a large, known-good battery such as a car starter battery. After running the test, navigate to this page and press the ENTER button, then trim the value down by 5mΩ. PE

Power Electronics series

By Andrew Levido

Power Electronics

Part 1: DC-DC Converters

Power electronics is a very broad term that describes circuits with the primary function of handling electrical energy. In this series of articles, we will explore this area, with practical examples. I will share some useful tools and techniques.

Power electronics is all around us – the rapid growth of renewable energy, electric vehicles and the near-ubiquity of switching power supplies means that a huge percentage of our electrical energy is generated, transmitted or consumed by power electronics.

Fundamentally, power electronics systems convert electrical energy from one form (a source) to another (a load). Consider a smartphone; there is power electronics in the charger, converting the mains to an isolated low voltage. There is more in the battery charging circuit within the phone, and yet more to provide the many low-voltage power rails necessary for its operation from the battery.

That battery is most likely a single LiPo cell, a type of lithium-ion cell, which will vary from around 4.2V when fully charged to around 3.3V when discharged. So the internal power supply electronics needs to be designed to convert that varying voltage to several fixed, regulated voltages for consistent performance of the various subsystems in the phone (processor, display, RF etc).

At the other end of the spectrum, there are the kilowatt and megawatt-scale power electronics systems controlling industrial processes including variable-speed motor drives, robotics and a whole range of other applications. Electric and hybrid vehicles also rely on power electronics for their battery charging and management systems, and of course, in the drivetrain.

This series will take a simlar approach to Ian Bell's *Circuit Surgery* column, with formulae and fairly detailed analysis. However, we aim to make this comprehensible to just about anyone interested in the subject, so we will try to avoid any mathematics beyond what is (or at least should be) taught in high school.

We will also make sure to provide examples and down-to-earth explanations. We want to make this discussion accessible, despite the complexity of the topic!

Breadth and depth

What makes power electronics especially interesting (and maybe a little bit daunting) is the enormous breadth and depth of the field. You can get an idea of its breadth by looking at Fig.1. This shows the scope of a typical power electronics system. It consists of the following subsystems:

- **Power switching:** this is the heart of any power electronics system. Power is switched by semiconductor switches such as Mosfets, IGBTs, diodes, thyristors and the like.
- **Input and output filters:** filters form an integral part of most power electronics systems, as we shall see. They can also help ameliorate some of the negative consequences of switching like electromagnetic interference (EMI) and poor power factor.
- **Switch drivers:** this might not seem very interesting, but power electronics switches can have demanding drive requirements. The control terminals of the switches are often floating at high voltages, or switching rapidly between different voltages. The driver circuits therefore often have to include bootstrap power supplies, level shifting and high-voltage isolation.
- **Control loops:** most power electronics circuits require some form of closed-loop control. This can be a simple voltage regulator in a DC-to-DC converter, or may involve multiple electrical or electromechanical sensors spread across a complex industrial machine.
- **Loads:** the power electronics designer has to have a good understanding of the load, its characteristics in operation and any feedback transducers that may be involved.

As well as covering this breadth, power electronics requires an ability to look at the system through different lenses at different times; from a wide-angle perspective, right down to a detailed microscope-level viewpoint.

For example, take capacitors. In this article, we will introduce a high-level analysis technique that assumes the average current through a capacitor is always zero. Later, we will have to zoom right in and look at the non-ideal behaviour of the same capacitor, including its equivalent series resistance (ESR) and the effects of voltage and frequency on its dielectric.

In the following article next month, we will consider capacitors from a complex impedance perspective.

It is really important to use the right analysis lens at the right time. If you go too detailed too soon, you can become hopelessly mired in unnecessary

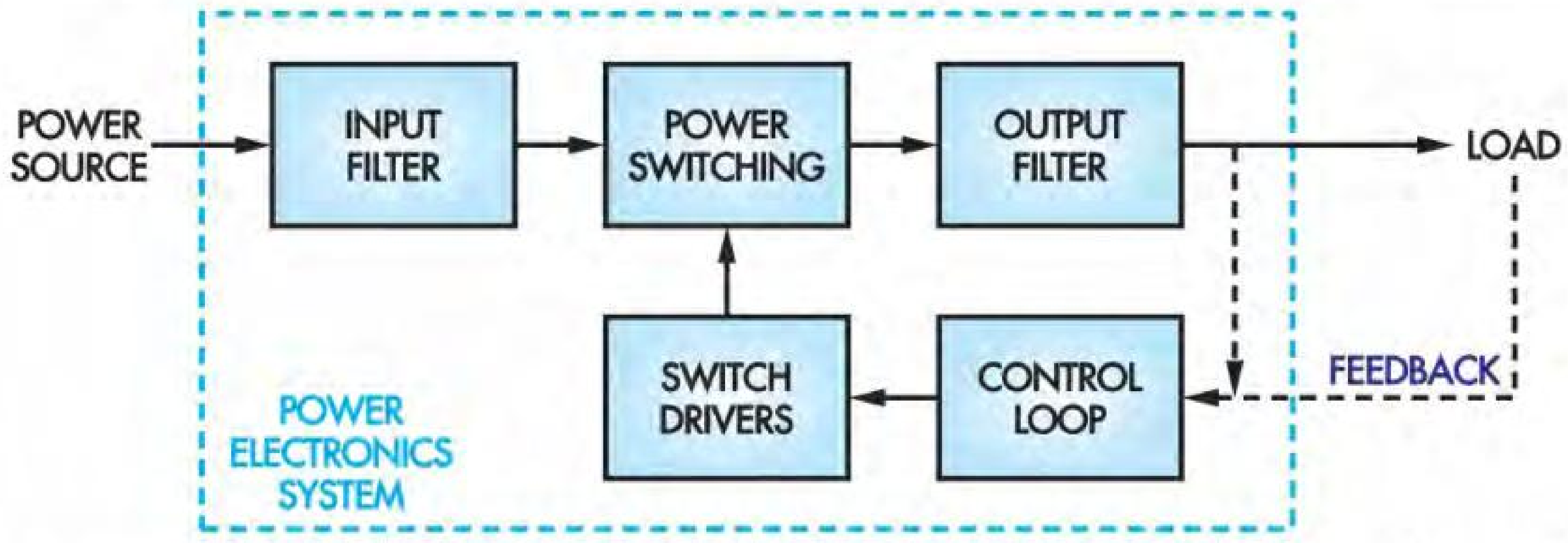


Fig.1: power electronics is a broad field, encompassing a knowledge of the load and the control loop as well as switching, drivers and filters.

complexity. Conversely, if you stay at too high a level for too long, you may over-simplify things and miss something important. I think this constant zooming in and out is one of the reasons why some people find power electronics difficult to grasp.

I am going to try to be very explicit about which lens we are using and when, to help dispel some of the complexity. To that end, we are going to zoom right out and start our analysis of DC-DC converter topologies with an annoyingly simple example.

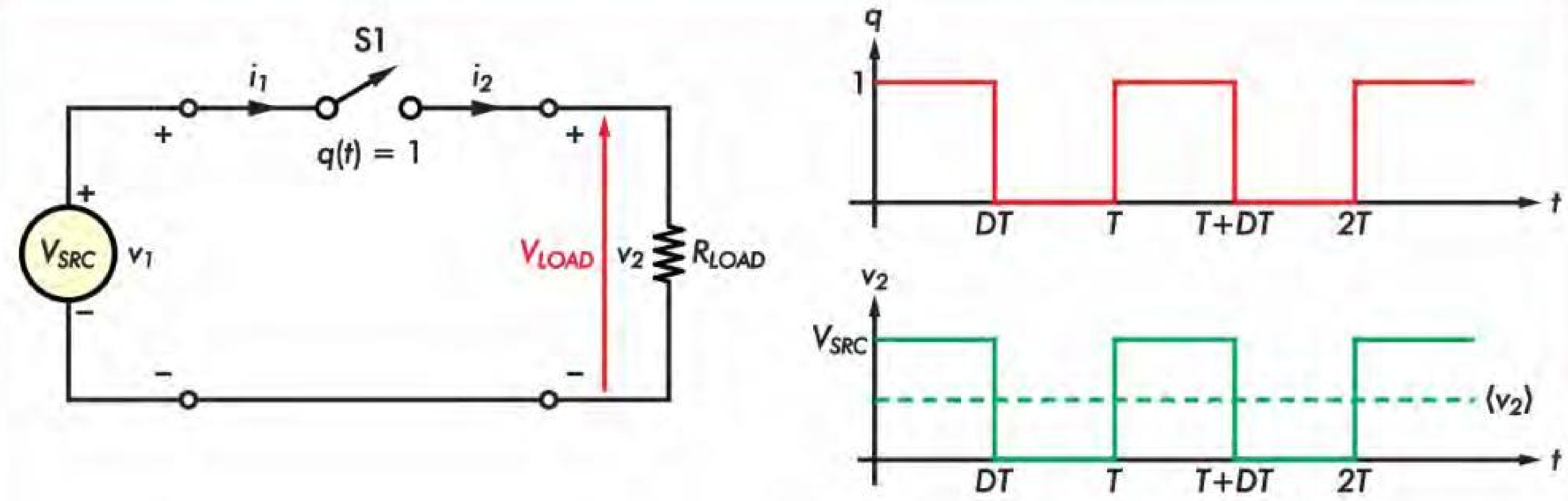


Fig.2: the simplest possible DC-DC switching converter. This is not very practical, but does reduce the average voltage efficiently. It's basically just pulse-width modulation (PWM).

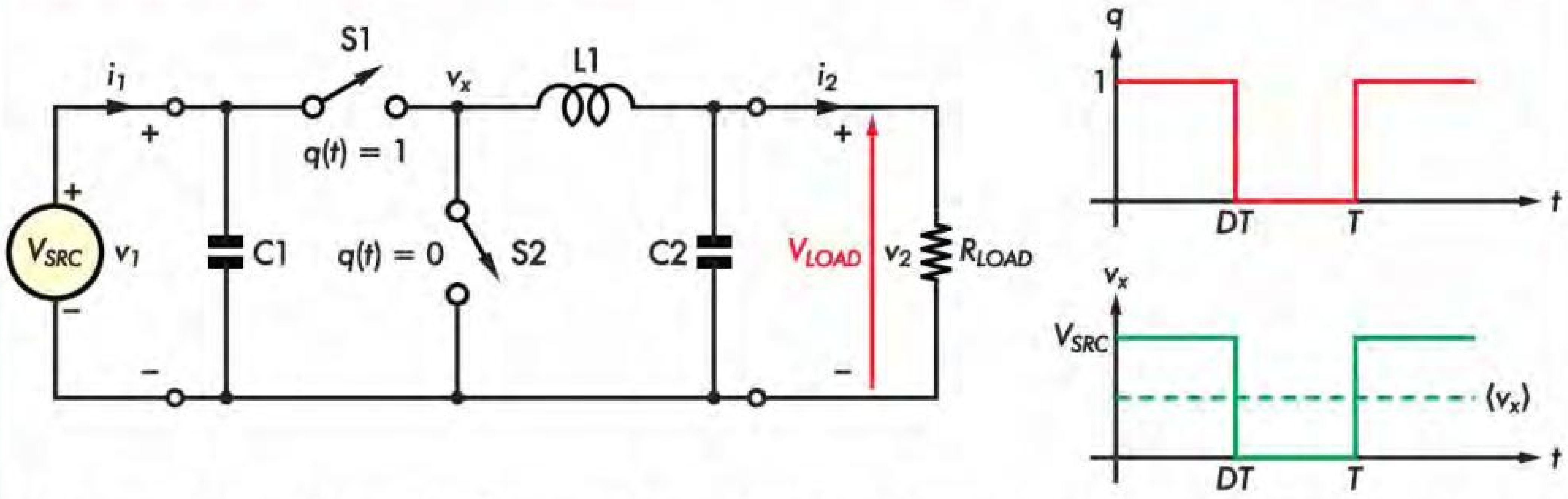


Fig.3: adding filters to the input and output of a simple switch produces a buck converter. The use of average value analysis allows us to work out the transfer function very easily.

A simple start

Let's start by assuming I want to build a switching DC-to-DC converter to reduce a source voltage, V_{SRC}, by 50% to power some load ($V_{LOAD} = ½V_{SRC}$). The simplest possible approach is shown in Fig.2.

If we operate switch S1 with a 50% on/off duty cycle (ie, on and off for equal periods), I think you would agree that the average voltage in the load (green dotted line) will be half that of the source voltage. The efficiency of the circuit will be 100%, since there are no lossy elements (much better than a linear regulator, which would be 50% efficient), so that is a win.

Of course, I am ignoring the obvious fact that the output voltage (green solid curve) will be far from a smooth DC voltage.

Before we move on to address this shortcoming, I want to take a minute to establish some conventions around the variables I will be using throughout this series.

- I will use lower-case variables to represent time-varying or AC quantities; for example, v_2 or q.
- I will use upper-case variables for fixed or DC quantities, like R_{LOAD} or V_{SRC}.
- I will use angle brackets to indicate an average value, like $\langle v_2 \rangle$. Obviously, it only makes sense to average time-varying values.

I am using $q(t)$ to describe a time-dependent control function that dictates the state of the switch. In the example of Fig.2, the control function (red trace) is a digital signal with a value of either 0 or 1. The label adjacent to S1 indicates that it is closed when $q(t) = 1$, and, by implication, is open when $q(t) = 0$.

The period of the control function is T, and it is on for a time DT, where D is the duty cycle, which can have any value between zero and one. The cycle period T is the inverse of the switching frequency, f_{SW}.

The next step is to add a filter to the circuit of Fig.2 to smooth the output. In Fig.3, we have added an LC low-pass filter (L1/C2) to the output, and a capacitor, C1, to the input. The input capacitor is unnecessary if the voltage source is perfect, but I have put it in because it is almost always required in real life, and it will come in handy later.

We will assume for now that the inductor and capacitor values are very large compared to the switching frequency so that the filtering is almost perfect. In this case, any capacitor voltage ripple or inductor current ripple are negligible.

The switch has now been expanded to two complimentary switches: S1, which is closed when $q(t) = 1$ as before, and S2, which is closed when $q(t) = 0$. The latter is necessary to provide a path for the inductor current (which can't change instantaneously) when S1 is opened.

Average value analysis

We are going to use a tool known as average value analysis to understand the transfer function (the relationship between the output and the input quantities) of this circuit. This kind of analysis is very useful for understanding the operation of a circuit at the highest level. We don't need to know any component values, or even the switching frequency, to complete this analysis.

Average value analysis requires our circuit to be in a condition known as 'periodic steady state' (PSS), which just means that the circuit is in a steady (unchanging) state at the macroscopic level. PSS therefore ignores transients like those that occur at start-up or where the load suddenly changes.

In this state, each switching period looks exactly the same, even though voltages and currents change during the cycle. If each period is identical, it follows that all voltage and current waveforms must start and end each period at the same value.

Something interesting happens to capacitors and inductors in PSS. Since the voltage across a capacitor starts and finishes each period at the same level, the average voltage across it must be constant. This implies that the average current through the capacitor must be zero.

A similar thing happens for inductors. In PSS, the average current through an inductor must be fixed. This means the average voltage across it must be zero. We can therefore write two equations that help us with average value analysis: $\langle i_C \rangle = 0$ and $\langle v_L \rangle = 0$. Be careful with this – it applies only to the average voltage and current during PSS.

Now we can perform an average value analysis on the circuit of Fig.3. If the average voltage across the inductor is zero, it should be apparent that $\langle v_x \rangle = \langle v_2 \rangle$. We can also calculate $\langle v_x \rangle$ from the green chart in Fig.3 and see that $\langle v_x \rangle = DV_{SRC}$. We can further see that, due to the perfect LC filter, $\langle v_2 \rangle$ is equal to V_{LOAD}. Putting these together, we find the circuit transfer function is $V_{LOAD} = DV_{SRC}$.

Since D must be between zero and one, V_{LOAD} must be equal to or lower than the input voltage. This circuit is a classic buck converter ('buck' meaning step-down in this context). Setting D to 0.5 gives us an output voltage of half the input voltage, as we wanted. This time, the output voltage is DC, ie, smooth.

If the average capacitor currents and inductor voltages are zero in PSS, there can be no average energy change, and therefore no power dissipation in the inductor or capacitors. This means that the converter input power and the output power must be equal ($I_{SRC} \bullet V_{SRC} = I_{LOAD} \bullet V_{LOAD}$).

Substituting the voltage transfer function derived above gives the current transfer function $I_{LOAD} = I_{SRC} \div D$.

Let's just pause for a second here and recap what we have done. Using nothing but the average value analysis rules (in PSS, $\langle i_C \rangle = 0$ and $\langle v_L \rangle = 0$), and a diagram of v_x, we have derived the voltage and current transfer functions for the classic buck converter. No fancy maths is required.

We have not had to worry about component values or the switching frequency – just the assumption that the LC filter time constant is much larger than the switching period T. Average value analysis is a powerful tool for understanding the basic function of switching converters.

While we are on a roll, let's look at what happens if we swap the source and load in our circuit. This is worthwhile to illustrate just how important the filters are in determining the operation of a power electronic converter.

A 'reverse' buck converter

Fig.4 shows our new converter. It is exactly the same as the buck converter, but I swapped the source and load. I have also switched the designations of the capacitors, voltages and currents so the subscript 1 is still associated with the source and subscript 2 with the load.

I have also swapped the control sense of the two switches. S1 is now on when $q(t) = 0$, while S2 is on when $q(t) = 1$. This aligns with the conventional way this type of converter is described, but does not change its operation.

Again, using average value analysis, we can see that if the average inductor voltage is zero, $\langle v_x \rangle$ must equal $\langle v_1 \rangle$, which is in turn equal to V_{SRC}. We can see from the graph of v_x (green curve) that its average $\langle v_x \rangle = (1 - D)V_{LOAD}$, so the transfer function of the converter must be $V_{LOAD} = V_{SRC} \div (1 - D)$.

We can use conservation of power as above to find that $I_{LOAD} = I_{SRC}(1 - D)$. Since D ranges between zero and one, the load voltage varies between V_{SRC} when $D = 0$ and approaches infinity as D approaches 1, so the output must be equal to or higher than the input. Therefore, this is a classic boost converter.

Additional topologies

We usually draw the boost converter with the source on the left, but I have done things this way to bring out the fact that the filter and switch arrangement is the same for both converters. Fig.5 summarises what we have just covered and adds several more topologies. I have used Mosfets and diodes in place of the switches to show how these circuits are usually implemented.

If you think about it, a diode can be considered a sort of passive switch, as it goes into and out of conduction depending on the voltage across it. While diodes are typically less efficient than Mosfets (due to their minimum forward voltage), they have the benefit of requiring no active control circuitry.

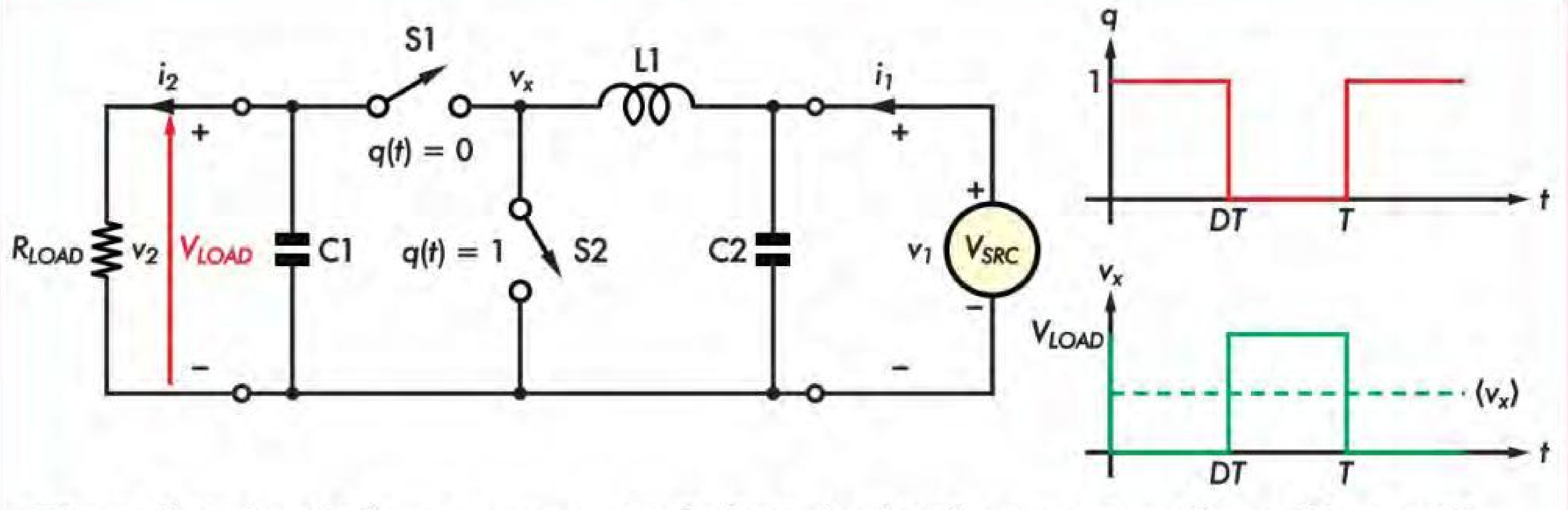


Fig.4: the classic boost converter is just the buck converter from Fig.3 with the source and load switched.

The first new converter is the buck-boost which, as its name suggests, should be able to produce a voltage above and below the input. This time, it should be obvious the average value of v_x must be zero, because the average value analysis rules dictate that the average inductor voltage must be zero.

However, we can see that when Q1 is on, v_x must equal V_{SRC}, and when it is off and the diode is conducting, v_x must equal V_{LOAD}. V_{LOAD} must thus be negative to make the average of $\langle v_x \rangle$ zero, as required by the average value analysis rules.

You can probably see from the plot of v_x that for the average to be zero, the area DV_{SRC} when Q is on must equal the area $-(1 - D)V_{LOAD}$, which is shown in the second equation. Rearranging gives the voltage transfer function shown in blue. As we expected, the output voltage is in the range of zero (when $D = 0$) and negative infinity ($D = 1$).

Of course, there will be a practical upper limit on the output voltage of these converters (usually on the order of a few times the input voltage), but for the purposes of this high-level analysis, it can be infinite.

The next converter we will look at is the Ćuk converter (pronounced "chook"). This was first presented by the American academic Slobodan Ćuk in 1976 (he was born in Belgrade, Yugoslavia, which is now part of Serbia). It is a departure from the previous converters, which use the inductor as the primary energy storage element.

The Ćuk converter uses a capacitor (C3 in Fig.5) as the main energy storage device, with the now-familiar LC filters on the input and output. Fortunately, we can use average value analysis to work out the transfer function in just the same way as we have before.

The upper waveform is the voltage at v_x and the lower waveform is the voltage v_y. When the Mosfet is on, v_x is zero and the left-hand end of C3 is grounded. This means that v_y must be equal to $-\langle v_C\rangle$. When the Mosfet is off, v_y must be zero, so the right-hand end of C3 is grounded, meaning v_x must be equal to $+\langle v_C\rangle$.

This leads to the first two equations that describe the average values of v_x and v_y, respectively.

The input and output filters mean that $\langle v_x\rangle$ is equal to V_{SRC}, and $\langle v_y\rangle$ is equal to V_{LOAD}, as shown in the next line of equations. Finally, we can combine these to produce a transfer function that is identical to that of the buck-boost converter, complete with voltage inversion.

Why would one use a Ćuk converter when it has the same transfer function as the simpler buck-boost converter? Firstly, the Mosfet is ground-referenced, making the drive circuit simpler. Secondly, it is possible to wind the two Ćuk inductors on the one core in such a way that the output ripple is dramatically reduced.

Finally, having LC filters on both input and output, the Ćuk converter can have lower EMI than other converter topologies.

The final type of converter I want to cover is the 'single ended primary inductor converter', generally abbreviated to SEPIC. This one also uses a capacitor as the energy transfer element. We'll analyse this in the same way as all the others.

When the Mosfet is on, v_x is zero and when it is off, v_x is the average capacitor voltage $\langle v_C\rangle$ plus the output voltage V_{LOAD}, as shown in the waveform. The average value of v_x is given by the first equation. In this converter, $\langle v_y\rangle$ must be zero due to inductor L2; therefore, $\langle v_x\rangle$ must equal $\langle v_C\rangle$.

We also know that, due to the input filter, $V_{SRC} = \langle v_x\rangle$, so we can re-write the first equation in terms of V_{SRC}, as shown in the third equation.

Finally, we can rearrange this to get the voltage transfer function, which turns out to be similar to the buck-boost and Ćuk converters, except not inverted. The SEPIC converter can therefore produce a positive voltage between zero and (in theory) infinity.

As you might imagine, there are many other variations on this theme, and their characteristics are not always obvious from just looking at the circuit. I hope that you have seen that average value analysis is a simple way to get to grips with switching converters.

Getting practical

Enough theory for now. I want to build the buck converter we have been discussing, to see what's involved and how closely reality matches the theory. I am also going to simulate it using a free circuit simulator called QSpice. To complete this design, we will have to zoom down into some of the detail, especially when it comes to capacitors.

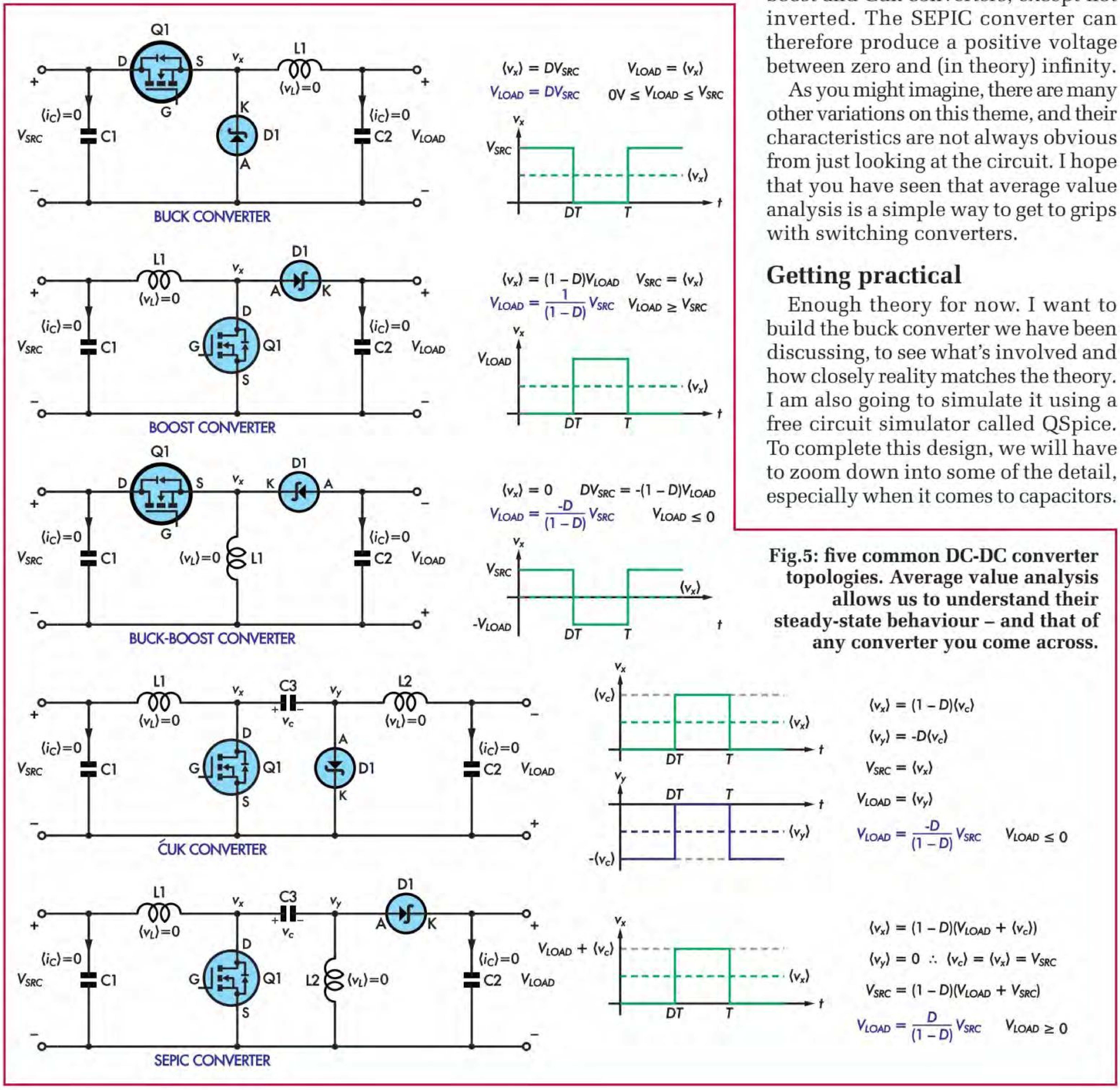

BUCK CONVERTER
Q1
D
S
G
vx
L1
⟨vL⟩=0
K
D1
A
⟨iC⟩=0
VSRC
C1
C2
VLOAD
⟨vx⟩ = DVSRC
VLOAD = DVSRC
VLOAD = ⟨vx⟩
0V ≤ VLOAD ≤ VSRC
vx
VSRC
⟨vx⟩
DT
T
t
BOOST CONVERTER
⟨vx⟩ = (1 – D)VLOAD
VSRC = ⟨vx⟩
VLOAD = 1/(1 – D) VSRC
VLOAD ≥ VSRC
VLOAD
BUCK-BOOST CONVERTER
⟨vx⟩ = 0
DVSRC = -(1 – D)VLOAD
VLOAD = -D/(1 – D) VSRC
VLOAD ≤ 0
-VLOAD
ĆUK CONVERTER
C3
vy
vc
L2
⟨vc⟩
-⟨vc⟩
⟨vy⟩
⟨vx⟩ = (1 – D)⟨vc⟩
⟨vy⟩ = -D⟨vc⟩
VSRC = ⟨vx⟩
VLOAD = ⟨vy⟩
VLOAD = -D/(1 – D) VSRC
VLOAD ≤ 0
SEPIC CONVERTER
VLOAD + ⟨vc⟩
⟨vx⟩ = (1 – D)(VLOAD + ⟨vc⟩)
⟨vy⟩ = 0 ∴ ⟨vc⟩ = ⟨vx⟩ = VSRC
VSRC = (1 – D)(VLOAD + VSRC)
VLOAD = D/(1 – D) VSRC
VLOAD ≥ 0

Fig.5: five common DC-DC converter topologies. Average value analysis allows us to understand their steady-state behaviour – and that of any converter you come across.

We will start with some specifications. The input voltage is nominally 12V, but let's allow for a range of 10V to 14V. We want an output voltage of 6.0V and a maximum output current of 600mA, so a 10Ω load. We can use the transfer function derived above to calculate the required duty cycle range: $0.43 \leq D \leq 0.6$.

I will use a TPS5410 DC-DC converter chip in this example. This is a pretty old chip, and not necessarily one I would recommend for new designs, but it has a couple of advantages for us. It has a fixed switching frequency of 500kHz and uses an external flyback diode. These make it a good match to the theoretical circuit, and allow us to measure the inductor current ripple fairly easily.

The circuit is shown in Fig.6. I have shown the connection of the internal Mosfet so you can see how it matches up with S1 in Fig.3. We will more-or-less follow the design procedure set out in the chip's data sheet, but I will take a bit of time to explain the formulas it provides so you can see how they are arrived at.

The capacitor C_{BOOT} is used by a bootstrap circuit within REG1 to create a drive voltage for the internal Mosfet that is a few volts higher than V_{SRC}. We will just use the manufacturer's recommended value of 10nF here.

The output voltage feedback comes via the voltage divider R1/R2. The V_{SENS} voltage on pin 4 should be 1.22V when the output voltage is at the desired level (6V in our case). If we let R2 = 10kΩ we can calculate that we need a value of 39.2kΩ for R2. This is very close to the standard value of 39kΩ, so we will use that.

I will use an MBRA130LT3 schottky diode for S2 (D1) since I have one on hand. This is a 1A, 30V fast diode, so should be fine for this application, since the peak current should be just a little over the load current of 600mA. That's it for the easy parts.

Input capacitor selection

The design process in the data sheet suggests that we start by selecting C1 to give the desired worst-case voltage ripple at the input.

Fig.7 shows the data sheet equation (at the bottom) and the circuit fragment that will help us understand it. The capacitor is shown together with its equivalent series resistance (ESR), since this will be significant in calculating the ripple.

The capacitor ripple voltage Δv_c is given by the top equation – the first part is the basic equation for the change in capacitor voltage as the current i_c is extracted, and the second part is the voltage across R_{ESR} from Ohm's law.

Fig.7 shows us that when S1 is closed, the capacitor discharge current i_c will be the net of the load current I_{LOAD}, less the charging current, I_{SRC}. The worst-case current occurs when $D = 0.5$ and $I_{SRC} = ½I_{LOAD}$. The capacitor current for the worst-case ripple must therefore be $½I_{LOAD}$.

The worst-case ripple voltage is therefore given by the second equation, which substitutes $½I_{LOAD}$ for i_c and $0.5T$ for Δt. This is almost identical to the data sheet formula. I have highlighted each term in a different colour so you can see how they are related.

The mysterious ¼ term in the data sheet formula is simply the $½I_{LOAD}$ factor and 0.5 worst-case duty cycle combined. f_{SW} is the switching period T shifted from the numerator to the denominator. The only real difference between the two equations is the missing $½I_{LOAD}$ factor in the ESR term. This will just make the ripple estimate a little bit higher, which is of little consequence.

Now we have to choose a capacitor with the appropriate value and ESR. This is not simple exercise when dealing with high-frequency circuits such as this. I want to use a multi-layer ceramic capacitor (MLCC) for its low ESR, in parallel with an aluminium electrolytic for bulk storage.

For the MLCC, I chose a 4.7µF 25V X7R model (Samsung CL21B475KAFNNNE), while for the electrolytic, I will use a 100µF 35V unit (Panasonic EEE-FP1V101AP), both of which I have in my parts bins.

It's complicated!

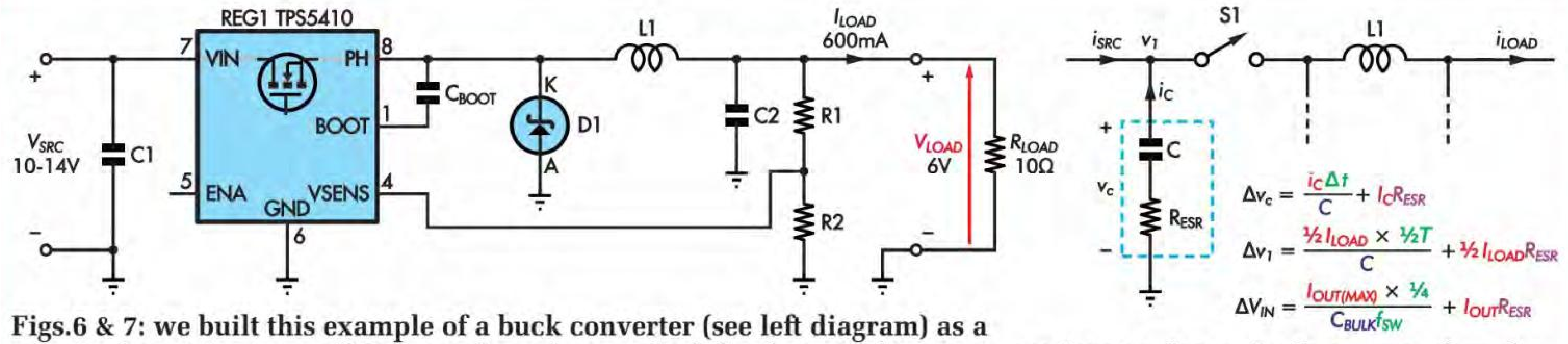
Now it is time to get out the microscope, because in high-frequency power electronics circuits (and indeed in many circuits), you can't necessarily take capacitors at face value. The chart at the top left of Fig.8 shows that, although our 4.7µF MLCC capacitor is rated for 25V, its nominal capacitance will fall by 70% (to 1.4µF) when biased with 12V.

This is a 'feature' of many MLCC dielectrics like X7R and X5R that you should be aware of. The capacitance also goes down with temperature and ageing, but not significantly in this application.

This is a good reason to choose MLCCs with a higher voltage rating than might seem necessary at first (or, in applications requiring lower capacitance, selecting a more stable dielectric like C0G/NP0).

The lower graph shows that, at 500kHz, the capacitor's ESR is about 4.5mΩ, which is very low, and one of the main reasons you see these capacitors in switch-mode circuits.

Fortunately, the capacitance of the electrolytic is relatively fixed with applied voltage and, according to its data sheet, it has an ESR of around 80mΩ at 100kHz. There is no data for higher frequencies, so we will assume this ESR holds at 500kHz.

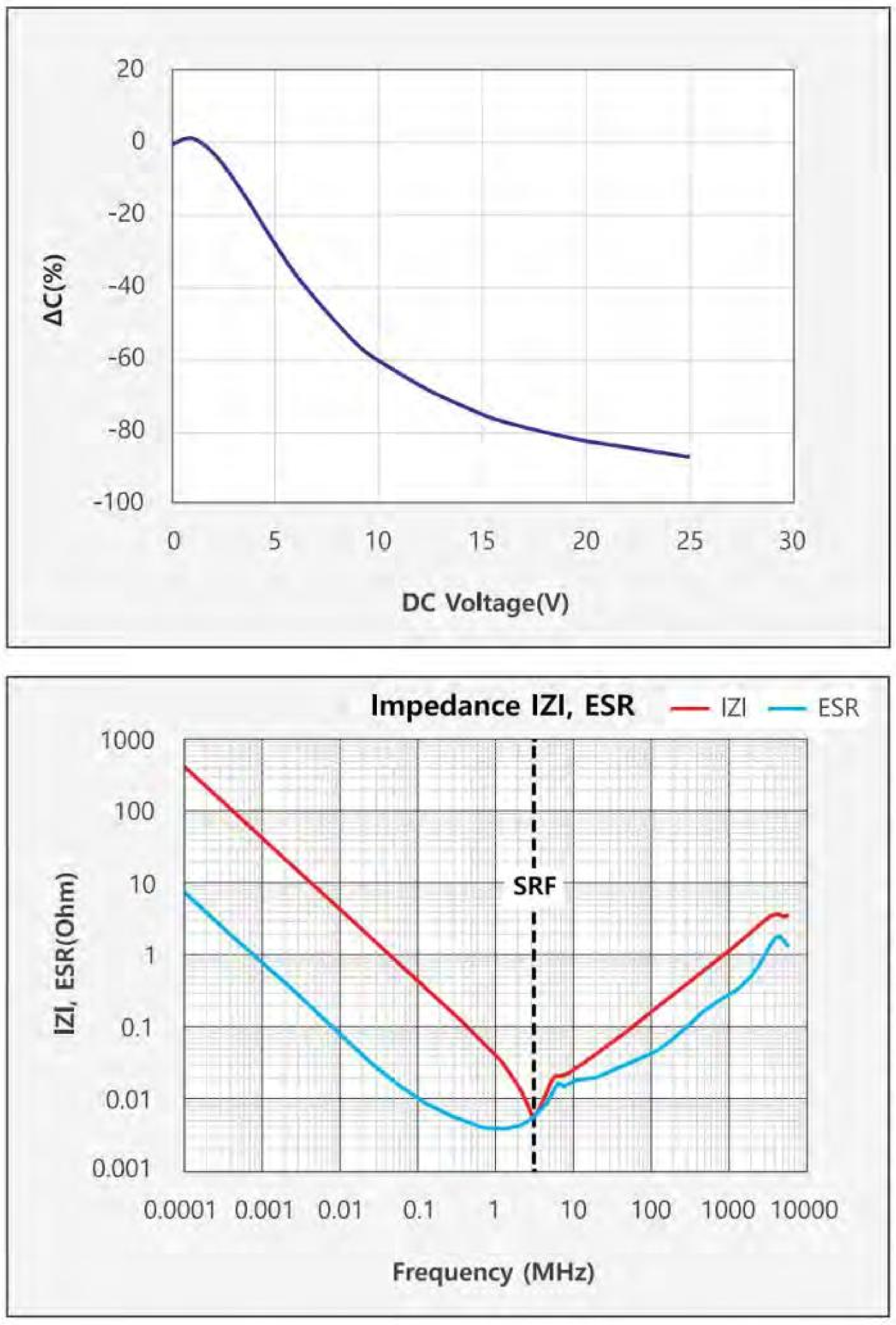


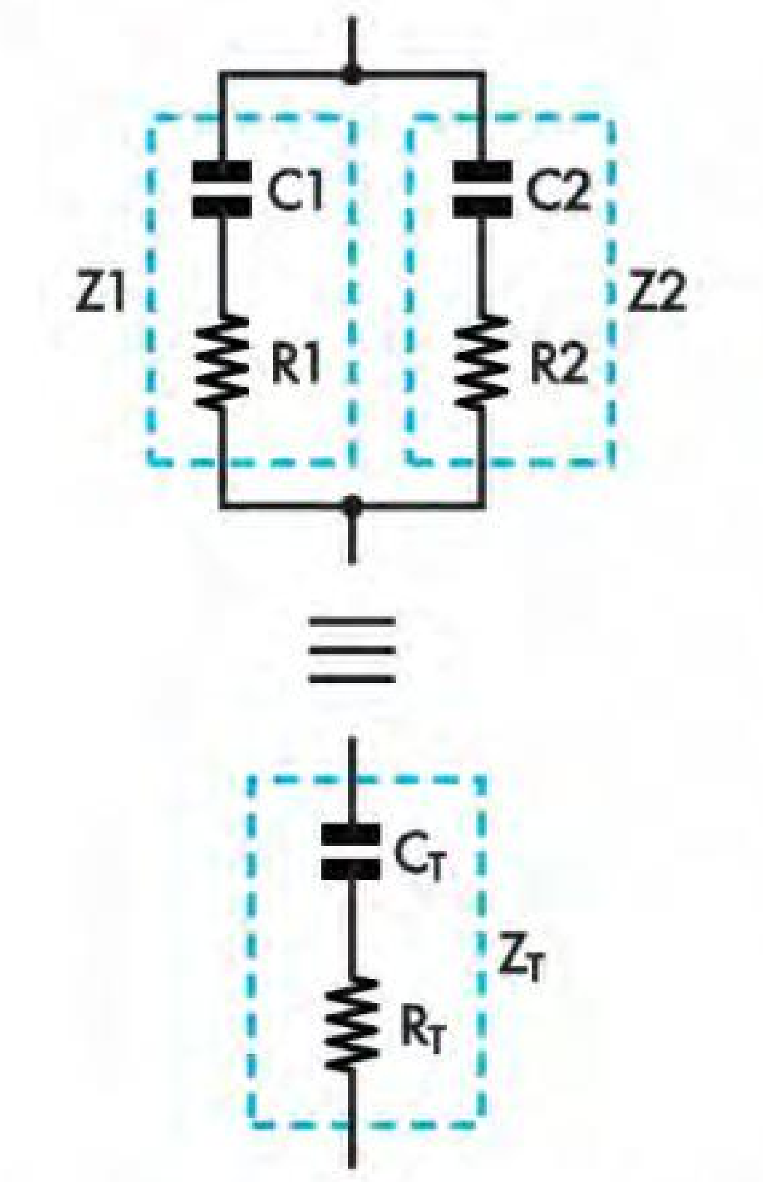
$$\Delta v_c = \frac{i_C \Delta t}{C} + I_C R_{ESR}$$

$$\Delta v_1 = \frac{½ I_{LOAD} \times ½T}{C} + ½ I_{LOAD} R_{ESR}$$

$$\Delta V_{IN} = \frac{I_{OUT(MAX)} \times ¼}{C_{BULK} f_{SW}} + I_{OUT} R_{ESR}$$

Figs.6 & 7: we built this example of a buck converter (see left diagram) as a practical exercise to see if the actual results matched the theory. The worst case input voltage ripple occurs when the duty cycle is 50% (see right diagram). The lower-most equation is copied from the data sheet, while those above it show how it was derived.





$$Z = R - j\frac{1}{2\pi fC} \qquad Z_T = \frac{1}{\frac{1}{Z1}\ \frac{1}{Z2}}$$

Fig.8: the plots on the left show that the capacitance and ESR of an MLCC can be very different from the nominal values under DC bias and at high frequency. The circuit and formulae above shows how we combine parallel impedances; for example, when paralleling capacitors with significant ESR.

Editor's note – be careful with such assumptions because wet electrolytic capacitors have physical limitations in response time.

Our two capacitors will be in parallel, but the diagram on the right of Fig.8 shows us why we can't assume the total capacitance will be the sum of the two capacitances, or that the total ESR will be the parallel combination of the two resistances.

Instead, we have to calculate the complex impedance of each capacitor/ESR combination (left equation), then calculate their parallel impedance (right equation) and decompose that to obtain the equivalent capacitor and resistance value.

I use a spreadsheet to do these (literally) complex calculations, and the results can be surprising and sometimes counter-intuitive. For example, my spreadsheet shows that the parallel combination we will be using will have an equivalent capacitance of 11.8µF and an ESR of 69mΩ at 500kHz.

Plugging these values into the input ripple formula suggests we can expect a peak-peak ripple voltage of about 46mV (67mV if you use the data sheet calculation, with its extra current in the ESR term).

Inductor selection

The next item we need to select is the inductor. The lower blue formula in Fig.9 is from the data sheet, and the circuit fragment shows us the simplified circuit when S2 is closed. It should be apparent that the voltage across the inductor is equal to the output voltage, V_{LOAD}. During this period, the inductor current will ramp down by some amount, Δi_L, that represents the peak-to-peak current ripple.

The first formula in the figure describes the general relationship between the rate-of-change of current in inductor and the voltage across it, arranged to make the inductance the subject. The second equation therefore shows the minimum inductance required to limit the current ripple to some value Δi_L. In this equation, v_L has been replaced by V_{LOAD} and Δt by $(1 - D)T$.

The data sheet formula is almost identical to this; the expression

$$(\mathbf{V_{IN(MAX)}} - \mathbf{V_{OUT}}) \div \mathbf{V_{IN(MAX)}}$$

evaluates to $1 - D$, and T moves to the denominator as f_{SW}. The current ripple is expressed as a factor (K_{IND}) of the load current. The random-looking 0.8 is there as a safety factor, to allow for inductor tolerance in the case you use an inductor with a value of exactly L_{MIN}.

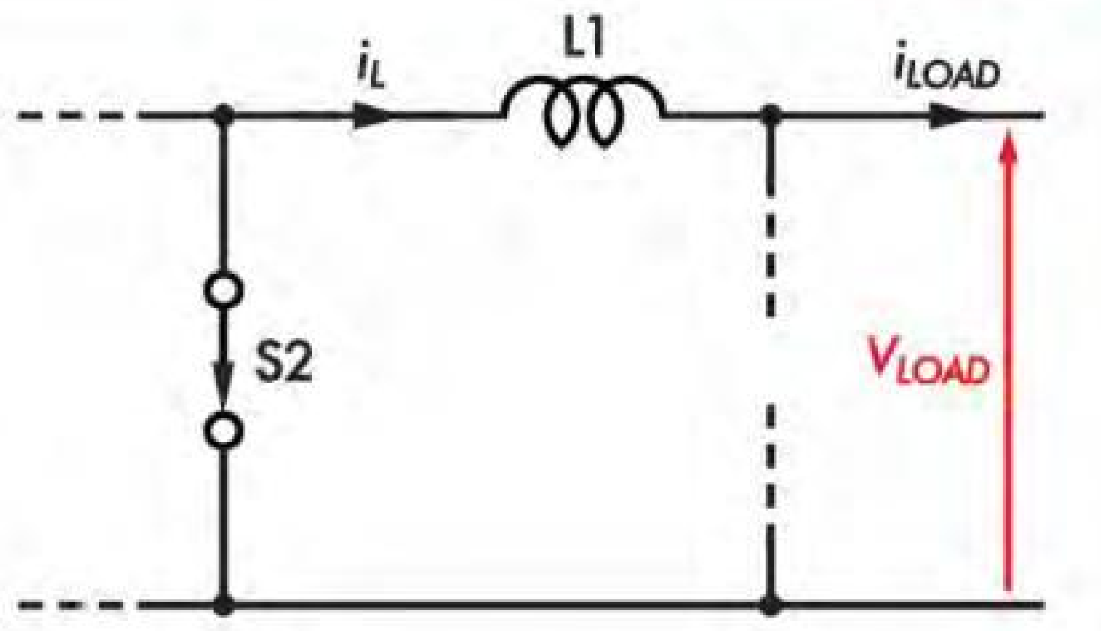
If we choose to have maximum ripple current of 100mA, we can calculate that we need an inductor with a value greater than 85µH. The nearest larger practical value is 100µH, so that is what we will use. If we back-calculate the ripple using this value, we get an expected peak-to-peak ripple current of 68mA.

The peak inductor (and diode) current will be the average load current plus half of the ripple current, or 634mA. I chose a VLS6045EX-101M inductor from TDK. This has a peak current rating of 1.1A and a DC resistance of 470mΩ (we want to keep that resistance low for good efficiency, as all the current flows through it).

Output capacitor selection

The capacitance of the output capacitor is not all that critical as far as output ripple is concerned – the output cap's ESR is usually the important parameter. In theory, you could keep adding output capacitance to make the ripple as low as you want, but this would cause problems with the performance of the voltage regulating control loop.

We will cover (some) control theory in the next article, but for now it is enough to know that the internal compensation in this particular controller requires the closed-loop crossover frequency to be in the range 3kHz < f_C < 30kHz.

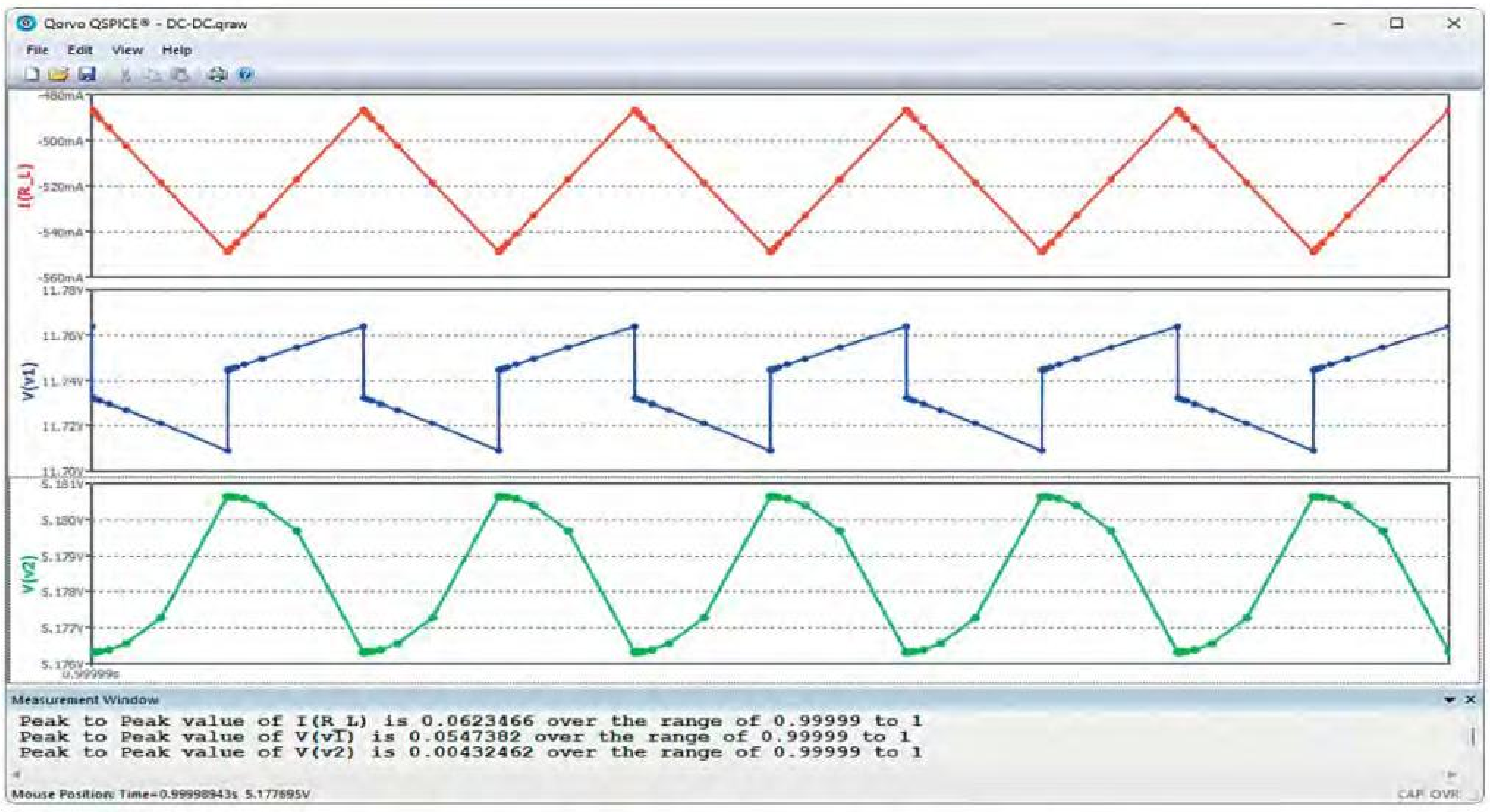


$$L = \frac{v_L \Delta t}{\Delta i_L}$$

$$L \geq \frac{V_{LOAD}(1 - D_{min})T}{\Delta i_{L(pp)}}$$

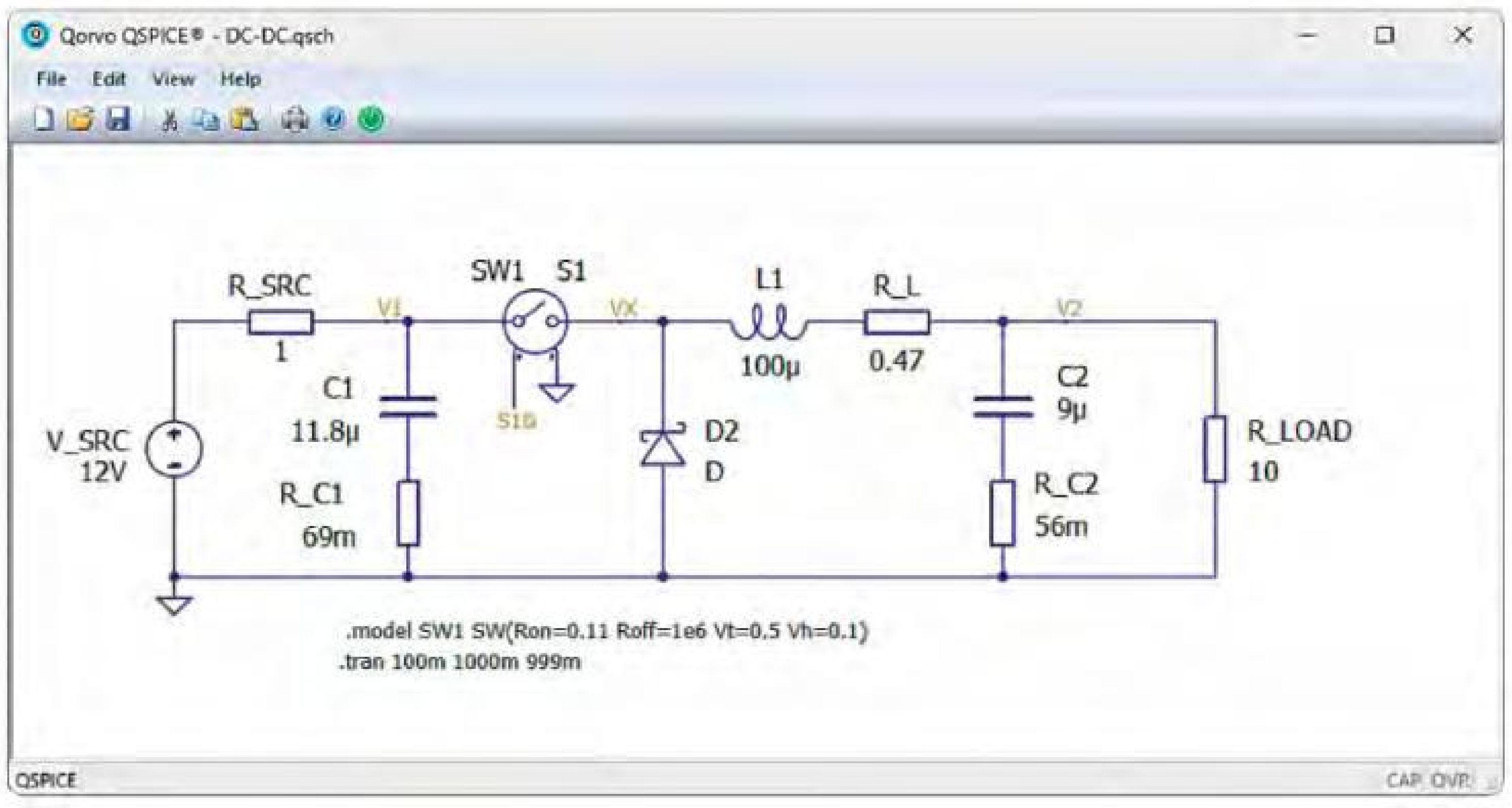
$$L_{MIN} = \frac{V_{OUT}(V_{IN(MAX)} - V_{OUT})}{0.8 \cdot V_{IN(MAX)} K_{IND} I_{OUT} F_{SW}}$$

Fig.9: the data sheet equation for peak to peak current ripple at the bottom of the figure is derived from the basic relationship between current and voltage in an inductor.



```
Peak to Peak value of I(R_L) is 0.0623466 over the range of 0.99999 to 1
Peak to Peak value of V(v1) is 0.0547382 over the range of 0.99999 to 1
Peak to Peak value of V(v2) is 0.00432462 over the range of 0.99999 to 1
```

Mouse Position: Time=0.99998943s 5.177695V



```
.model SW1 SW(Ron=0.11 Roff=1e6 Vt=0.5 Vh=0.1)
.tran 100m 1000m 999m
```

Fig.10: simulation of the circuit (left) yields results (above) that are very consistent with the calculated values.

The closed-loop crossover frequency must also be lower than the frequency of the 'pole' formed by the output capacitor and its ESR. The closed-loop crossover frequency is the frequency at which the closed-loop gain falls to zero.

The manufacturer helpfully provides an expression that relates f_C to the output filter corner frequency, f_{LC}. Given we know the inductance, we can determine that C2 must be between 165.6µF and 31.4µF (at the loop crossover frequency, not necessarily at 500kHz).

We can therefore use the same 4.7µF/100µF pair as we did for the input, since with 6V DC bias and the 4.5kHz crossover frequency the, parallel capacitance is about 102.5µF and the ESR is 76mΩ. This will have an ESR pole at 22kHz, much higher than the crossover frequency, so we should have a stable control loop.

Notice that we have used the loop frequency to calculate the capacitance and ESR for loop stability, and that we will use the switching frequency to calculate them for ripple below. This is a great example of why power electronics can seem confusing – the same capacitor has different apparent values depending on the type of analysis we are performing.

The output ripple is then more-or-less trivial to calculate, as it is just the output cap's ESR multiplied by the peak-to-peak ripple current. At 500kHz with 6V bias, the tables and the spreadsheet tell us that total capacitance will be 9µF and the ESR 56mΩ. With a current ripple of 68mA, we should see about 3.8mV of peak-to-peak voltage ripple at the output.

That much ripple is not likely to be a problem for whatever it is driving. With additional filtering, it could be reduced well below 1mV.

Simulation

Simulation is a powerful tool we can use to analyse the behaviour of power electronics circuits. There are a few very capable free simulators available. I have been using QSpice recently, as its user interface (UI) seems to be better than some of its competitors.

Like everything in power electronics, the trick to good simulation is to work out what is important and what can be simplified or ignored. Fig.10 shows where I landed. A subcircuit (not shown) produces a control signal for the switch with a period of 2µs and a duty cycle of 0.5.

I have given the input voltage a source impedance of 1Ω, which is reasonable, as perfect voltage sources don't exist in reality. The input and output capacitances are represented by capacitors with series ESR resistances using the values that we just calculated. I have added the inductor's series resistance for completeness.

S2 is represented by a generic schottky diode, and S1 by a switch with an on-resistance of 110mΩ, like the Mosfet in the TPS5410.

The ".tran" directive tells QSpice to perform a transient analysis over one second, but to display only the last millisecond. This is necessary because, when the converter starts from zero, it takes a little while to reach a steady state.

The simulation output at the top of Fig.10 shows the inductor current at the top, the input voltage ripple in the centre and the output voltage ripple at the bottom. QSpice offers some handy utilities, including one to measure the peak-to-peak values of the displayed waveforms. You can see that the current ripple is 62mA, the input ripple is 55mV and the output ripple is 4.3mV.

This accords pretty well with the calculated values (68mA for current ripple, 46mV or 67mV for input ripple and 3.8mV for output ripple). This is not surprising, since the simulation is using the same inputs as the calculations. The proof of the pudding would be to build the circuit and see how it performs.

Test results

I did just that, and the measured results are shown in Fig.11. The first oscilloscope trace is the current through the diode (measured across a 0.5Ω series resistor). This is a lot easier than measuring the inductor current directly. The ripple is derived from the slope on the top of the waveform. The measured ripple current is around 64mA, in line with the calculated value of 68mA.

The input ripple (centre) current looks a lot like that in the simulation, but you can see some ringing on the waveform. This is caused by stray inductance in the circuit (for example, in the leads to the power supply) resonating with the input capacitance. The scale is 20mV/division, so the peak-to-peak ripple is in the region of 45mV, again very close to the calculated value of 46mV.

Finally, the output ripple is roughly the same shape as the simulation with the exception of switching spikes. These are due to the fast transitions of v_x making their way through the inductor's stray parallel capacitance. Nevertheless, the amplitude of the underlying ripple is about 4mV peak-to-peak, bang on the calculated 3.8mV value.

If you want to eliminate the switching spikes, you can add a secondary LC filter; sometimes a ferrite bead is all it takes. But keep in mind that it may (likely will) affect transient regulation.

I should point out that making these measurements is not trivial. If you were to use a normal 'scope probe with its 150mm-long ground clip, you would see a whole lot of noise superimposed on these signals. Instead, I used a piece of thin 50Ω coax (RG316) with a BNC connector on one end.

The screen on the other end is stripped back about 10mm, with the core and screen connected directly across the capacitor or resistor whose voltage is being measured.

For signals that never exceed a volt or so, like the voltage across the 0.5Ω resistor used to measure the diode current, you can connect the coax directly to a scope with the 50Ω terminator enabled. You can't do this for higher voltage signals, such as when measuring the input or output ripple, since the 50Ω terminator is usually not rated for more than a few volts.

You may be able to use the normal high-impedance input and AC coupling, but I use a home-made power rail probe to eliminate the DC offset, allowing me to safely use the 50Ω input with its better noise performance. That device is described in a separate project in this issue, starting on page 47.

Conclusion

This introduction has demonstrated that power electronics is a field that requires the designer to shift back and forth between high-level circuit analysis and the minutiae of component behaviour. This is what makes it endlessly fascinating to me.

Next month, we will look in detail at what is involved in designing the control loop of DC-to-DC converters like this one. Not understanding such control loops is probably the #1 problem that people have implementing such DC/DC converters! PE

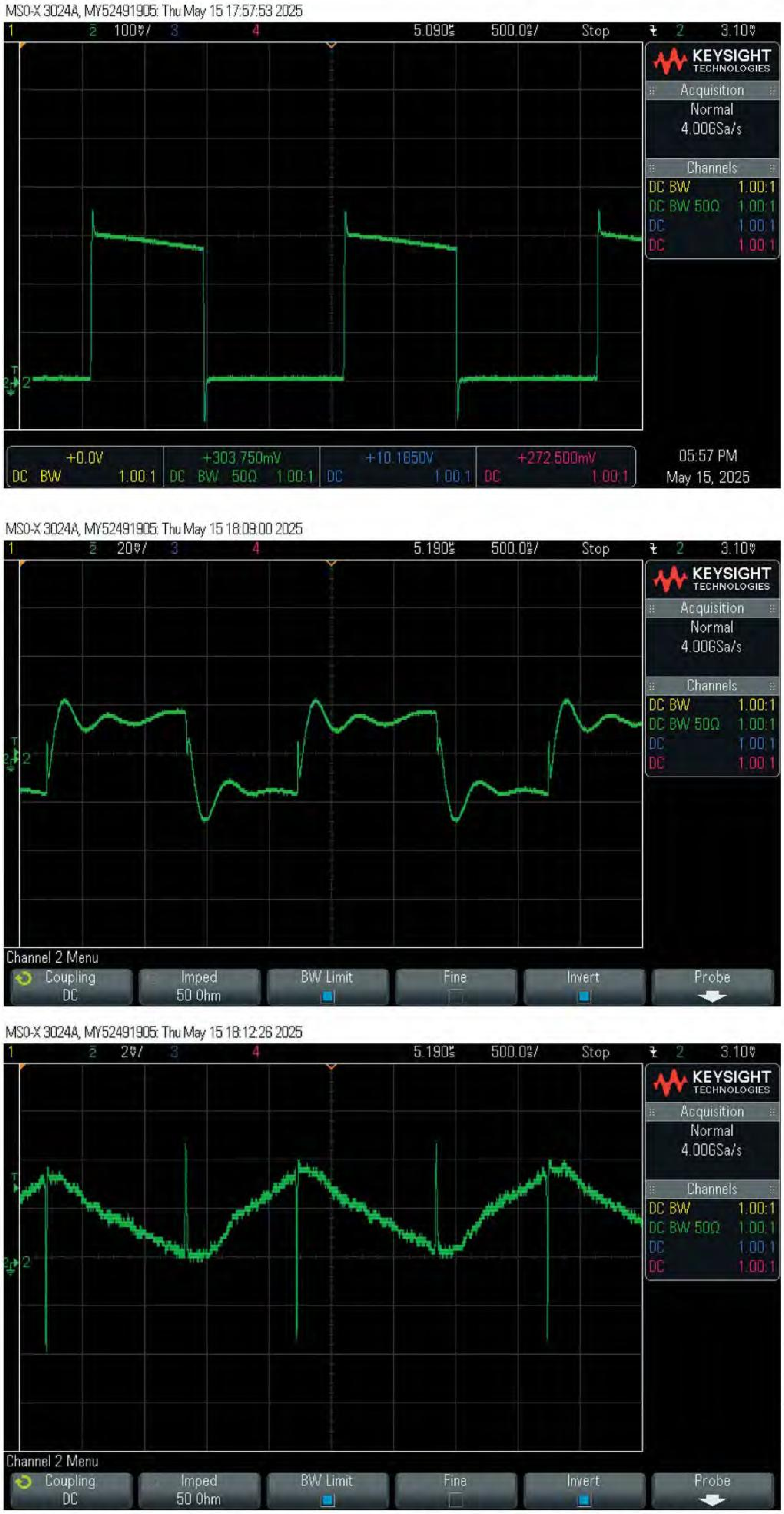

MSO-X 3024A, MY52491905: Thu May 15 17:57:53 2025
100mV/
5.090s
500.0ns/
Stop
3.10V
KEYSIGHT TECHNOLOGIES
Acquisition
Normal
4.00GSa/s
Channels
DC BW 1.00:1
DC BW 50Ω 1.00:1
DC 1.00:1
DC 1.00:1
+0.0V
+303.750mV
+10.1850V
+272.500mV
05:57 PM
May 15, 2025
MSO-X 3024A, MY52491905: Thu May 15 18:09:00 2025
20mV/
5.190s
Channel 2 Menu
Coupling DC
Imped 50 Ohm
BW Limit
Fine
Invert
Probe
MSO-X 3024A, MY52491905: Thu May 15 18:12:26 2025
2mV/

Fig.11: I built the circuit and measured the diode current (top), input voltage ripple (centre) and output voltage ripple (bottom). The results are very close to the calculated and simulated values.

Subscribe to SILICON CHIP

SILICON CHIP
2026
NZ $14.90
The history of intel
from the 1101 SRAM chip and beyond
DCC Remote Controller
run multiple trains through one DCC Base Station, with up to five Remote Controllers

Silicon Chip is one of the best DIY electronics magazines in the world. Each month is filled with a variety of projects that you can build yourself, along with features on a wide range of topics from in-depth electronics articles to general tech overviews.

	Print	Combined	Online
6 months	$105 ≈£54 ≈€62	$115 ≈£59.50 ≈€68	$50 ≈£26 ≈€30
1 year	$200 ≈£103 ≈€119	$220 ≈£114 ≈€131	$95 ≈£49 ≈€57
2 years	$390 ≈£202 ≈€232	$425 ≈£220 ≈€253	$185 ≈£96 ≈€110

- *Combined subscriptions include both the printed magazine and online access.*
- *Prices are in Australian dollars (AUD) unless otherwise specified.*
- *Prices listed are for everywhere in the world except Australia and New Zealand. View **siliconchip.com.au/Shop/SubRates** for a full list of current prices.*
- *Price estimates are just that and should only be used as a guide. We can only take payment in the form of Australian dollars.*

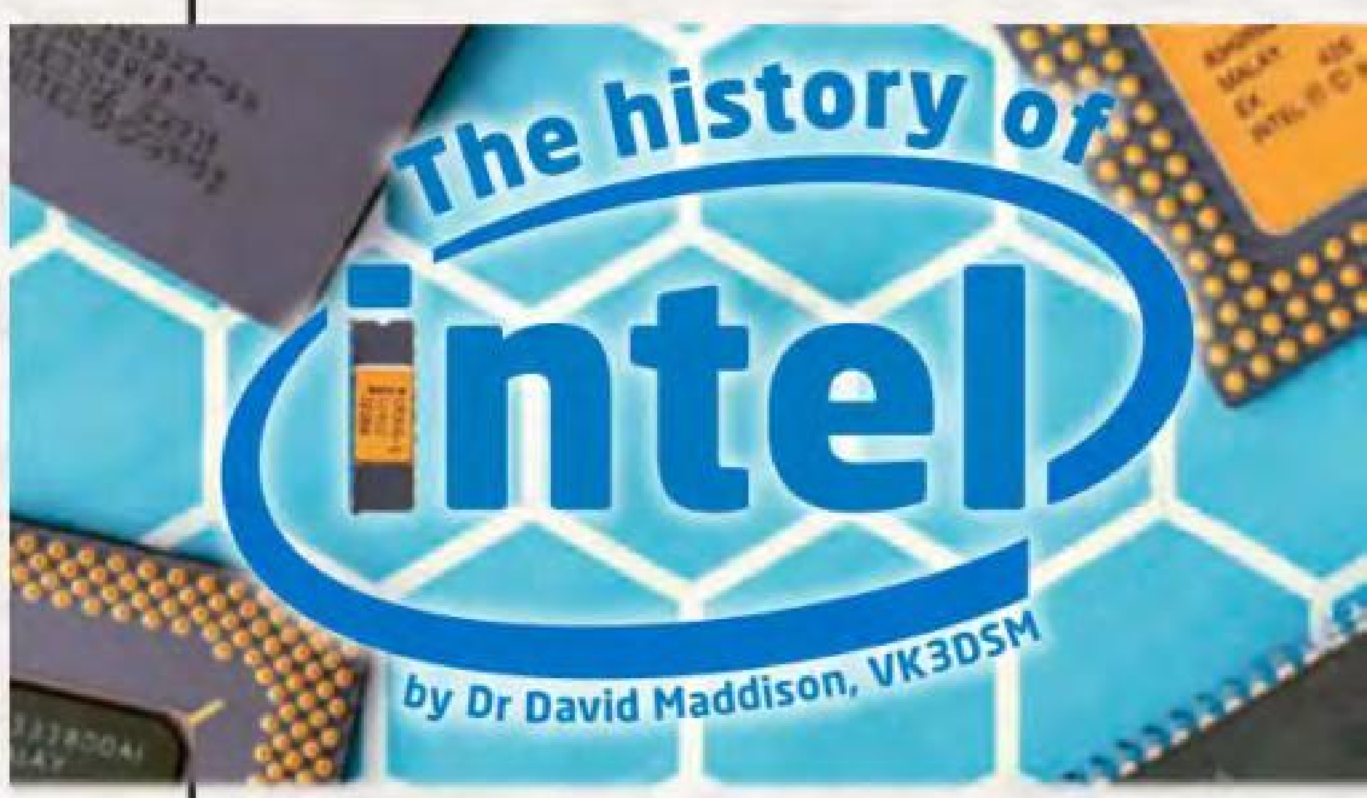
Try our Online Subscription – now with PDF downloads!

An online issue is perfect for those who don't want too much clutter around the house and is the same price worldwide. Issues can be viewed online, or downloaded as a PDF.

Published in Silicon Chip

The Internet Radio; Feb-Mar 2026

Mains Hum Notch Filter; March 2026


The history of
intel
by Dr David Maddison, VK3DSM

The History of Intel; Feb-Apr 2025

To start your subscription go to siliconchip.com.au/Shop/Subscribe

Power Rail Probe

This is one of those pieces of test equipment that you don't really need, until you do. It allows the measurement and evaluation of ripple, switching noise and transients riding on DC supply rails.

Project by Andrew Levido

INPUT
50 kΩ
Maximum
±50 V
OUTPUT
50 Ω
LOAD
Power Rail
Probe
POWER
Charge
OFF - ON
±25 V
P9058
CHARGE
Charge

High-performance commercial power rails probes are available, but they cost many thousands of dollars, putting them out of reach for most hobbyists. This project proves it does not have to be that way. The probe described here offers good performance for less than £50 in parts.

Passive oscilloscope probes are not really suited to looking at the ripple, switching noise and transients that can occur on power rails, especially those produced by switching converters. These are usually millivolt-level signals that are riding on top of a comparatively high DC voltage.

Also, there is usually a lot of radiated noise that can be picked up by a standard passive probe with its 150mm-long ground clip wire.

You will have to switch your oscilloscope to AC coupling to eliminate the DC offset to get the vertical resolution necessary to see the ripple and noise. This is fine if you are only interested in high-frequency artefacts, but no good for transients with time constants in the milliseconds range, like those you might encounter with a step change in load.

AC-coupling introduces a high-pass filter with a cutoff frequency in the 1-10Hz range into the signal path, which means the low-frequency and DC components will not be displayed accurately.

You may be able to use DC coupling if your scope's offset control has sufficient range. However, on the millivolt ranges, the offset is typically limited to a volt or two, so you likely won't be able to get the trace on the screen at all.

A power rail probe sits between the power rail being measured and the oscilloscope. It typically has an input impedance of 50kΩ, so it does not load the power rail too much, and an output designed to connect to a scope input in 50Ω impedance mode. The probe allows the DC offset to be removed but preserves the bandwidth from DC all the way to the upper bandwidth of the scope.

If your scope does not have a 50Ω termination option, you could use a separate terminator (for example, the Amphenol 112667).

Commercial models typically offer bandwidth up to the GHz range, can offset ±25V DC and can handle active signals in the range of ±1V. They usually attenuate the signal slightly, but this attenuation is known – and importantly – is constant across from DC to the upper bandwidth limit.

The Power Rail Probe described here meets most of those specifications. It has a DC input impedance of around 50kΩ and can offset power rails up to ±25V. It can handle a signal amplitude of at least ±1V and has a nominal attenuation of -1.7dB ±0.3dB from DC to at least 100MHz. The actual limit is almost certainly a fair bit higher, but that is how high I can measure with my test equipment.

The whole thing is built into a small plastic case. It is powered by a lithium polymer cell that can be recharged via a USB-C power source.

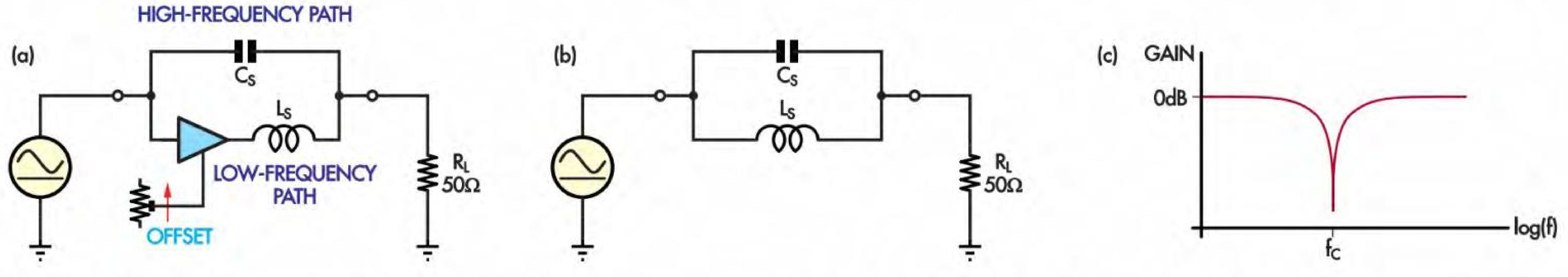
Design

The block diagram of the Probe is shown in Fig.1(a). There are two signal paths in parallel: a low-frequency path, where the DC offset is applied, and a high-frequency path that keeps the shape of the waveform intact. This architecture is necessary because the offset circuitry uses op amps and is therefore limited in bandwidth.

We will come back to the design of the offset circuitry later, but first we will focus on the two signal paths.

If we assumed the offset is zero and the buffer is perfect, the circuit reduces to the LC parallel topology shown in Fig.1(b). I have added the 50Ω load presented by the oscilloscope and shown a voltage source for clarity.

You will probably identify this circuit as a classic LC notch filter. At the resonant frequency of $1 \div (2\pi\sqrt{LC})$, the



Figs.1(a) - (c): the Power Rail Probe has a low-frequency signal path for the offset and a parallel high-frequency path, so it has very high bandwidth. The two paths would form an LC notch filter unless we introduce some resistance to lower the Q.

LiPo 602535
3.7V 400mAh

The PCB is sparsely populated & none of the devices are overly difficult to solder.

LC network will appear to have infinite impedance, so the response will have a distinct notch as shown in Fig.1(c). The steepness and depth of the notch is dictated by the Q of the filter.

We obviously don't want such a dip in our frequency response, so we need to introduce some resistance into the circuit to lower the Q.

We can do this by inserting resistors R_S in series with the inductor and capacitor, as shown in Fig.2(a). I have added the 'S' subscript to the inductor and capacitor values for reasons that will soon become apparent. These resistors should be equal in value to keep the attenuation constant at the extremes.

For example, at very low frequencies (or DC), the capacitor will present an infinite impedance, and the inductor will present a zero impedance, so the attenuation will be given by the voltage divider formed by R_S and R_L. At very high frequencies, the inductor will appear to have infinite impedance and the capacitor zero impedance, so the attenuation will again be $R_L \div (R_S + R_L)$.

Series-parallel equivalent circuit

It is hard to visualise what happens at the resonant frequency, since the capacitor and inductor are no longer in parallel. However, we can take advantage of a special property of complex impedances, called series-parallel equivalency. This just states that at any given frequency, there is a parallel and series combination of elements that behave identically when viewed from the terminals.

Fig.2(b) shows us what the network would look like after transformation. This means that there is a parallel R_P/C_P circuit that behaves exactly the same as the series R_S/C_S circuit in the high-pass branch, and a parallel R_P/L_P circuit that behaves exactly the same as the series R_S/L_S circuit in the low frequency branch at the resonant frequency.

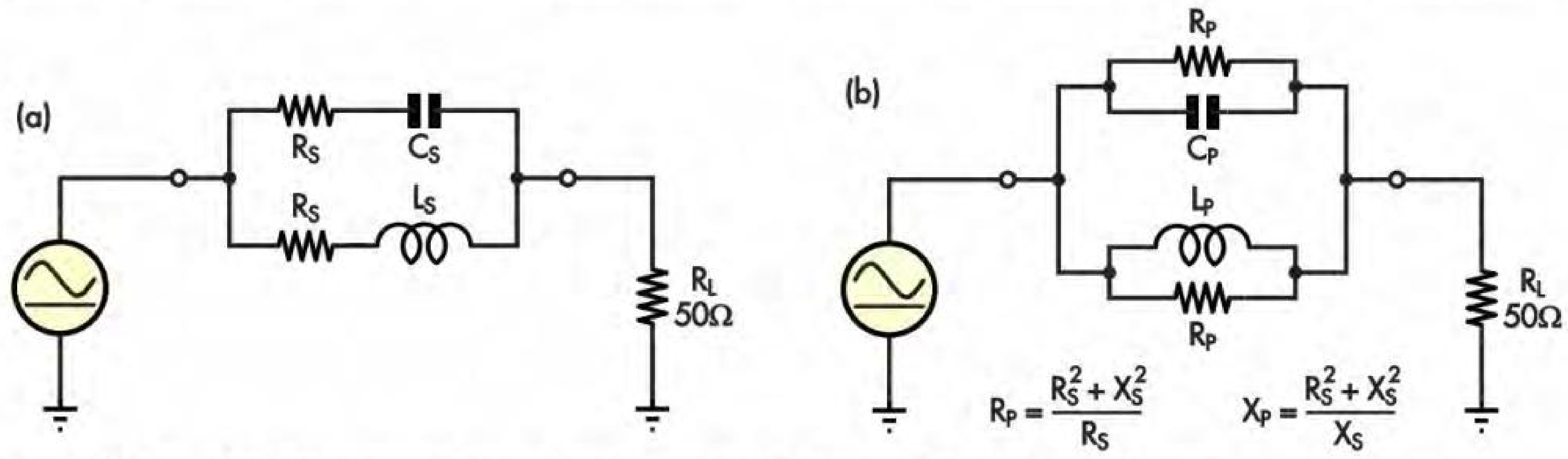
The component values in the parallel circuits will differ from those in the series circuits, but the behaviour will be the same. The formula in that figure shows how the parallel and series impedances are related.

With this transformation, the inductor and capacitor are now in parallel, so will have an infinite impedance at the resonant frequency. The impedance of the two paths at this frequency will therefore be determined by the two resistors in parallel (R_P).

To keep the attenuation at the resonant frequency the same as at high and low frequency cases, we need each R_P to be twice the value of R_S. Substituting this relationship into the equation for R_P in Fig.2, we can see that R_S must be equal to X_C or X_L (which will be identical to each other at the resonant frequency).

Using either one of these, plus the expression above for the resonant frequency, gives us the result that, for a flat response, R_S should be equal to $\sqrt{L \div C}$.

I chose R_S to be 10Ω to give an attenuation of 1.2 (around -1.7dB), in line with the commercial units. This means the inductance should be 100 times the capacitance (in terms of henries and farads), so I chose 10μH and 100nF – both readily available values. These values give a crossover frequency of around 159kHz.


(a)
R_S
C_S
R_S
L_S
R_L 50Ω
(b)
R_P
C_P
L_P
R_P
R_L 50Ω
$R_P = \frac{R_S^2 + X_S^2}{R_S}$
$X_P = \frac{R_S^2 + X_S^2}{X_S}$

Figs.2(a) & (b): the resistors in series with the inductor and capacitor form voltage dividers with the load resistance at very high and very low frequencies. This parallel equivalent circuit (b) behaves identically to (a) at any given frequency if the values are chosen appropriately. This allows us to calculate component values for a flat frequency response.

The low-frequency path

Fig.3 shows the full circuit of the Power Rail Probe. The 'ground' of the main signal path circuit (the horizontal line across the middle) is produced by op amp IC1c and the divider at its input. It settles at half the battery (cell) voltage, around 1.85V. The power supply for the op amps is therefore between ±2.1V and ±1.8V depending on the cell's state of charge.

Op amp IC1a forms an inverting, summing amplifier which adds the input voltage (via a 51kΩ resistor) with an offset voltage derived from potentiometer VR1. The input voltage is amplified by a factor of -1 (ie, inverted) due to the op amp's feedback resistor also being 51kΩ.

The ±1.8V present at the wiper of VR1 is amplified by -15.5, offsetting the input voltage by up to ±27V (or more if the battery voltage is higher). The second op amp, IC1d, is configured as an inverting buffer to flip the signal back to the right sense.

I have used a potentiometer with a mechanical detent and centre tap that is connected to the virtual ground. This makes the zero-offset point very easy to find. This is helpful because it is very easy to lose the trace on the millivolt range if the pot can shift the voltage by ±25V. An easy-to-find zero point makes it much easier to get the trace back on screen. That said, a standard three-terminal pot would work just fine.

The choice of op amps is quite important for the proper operation of the circuit. For once, we don't care too much about input offset voltages,

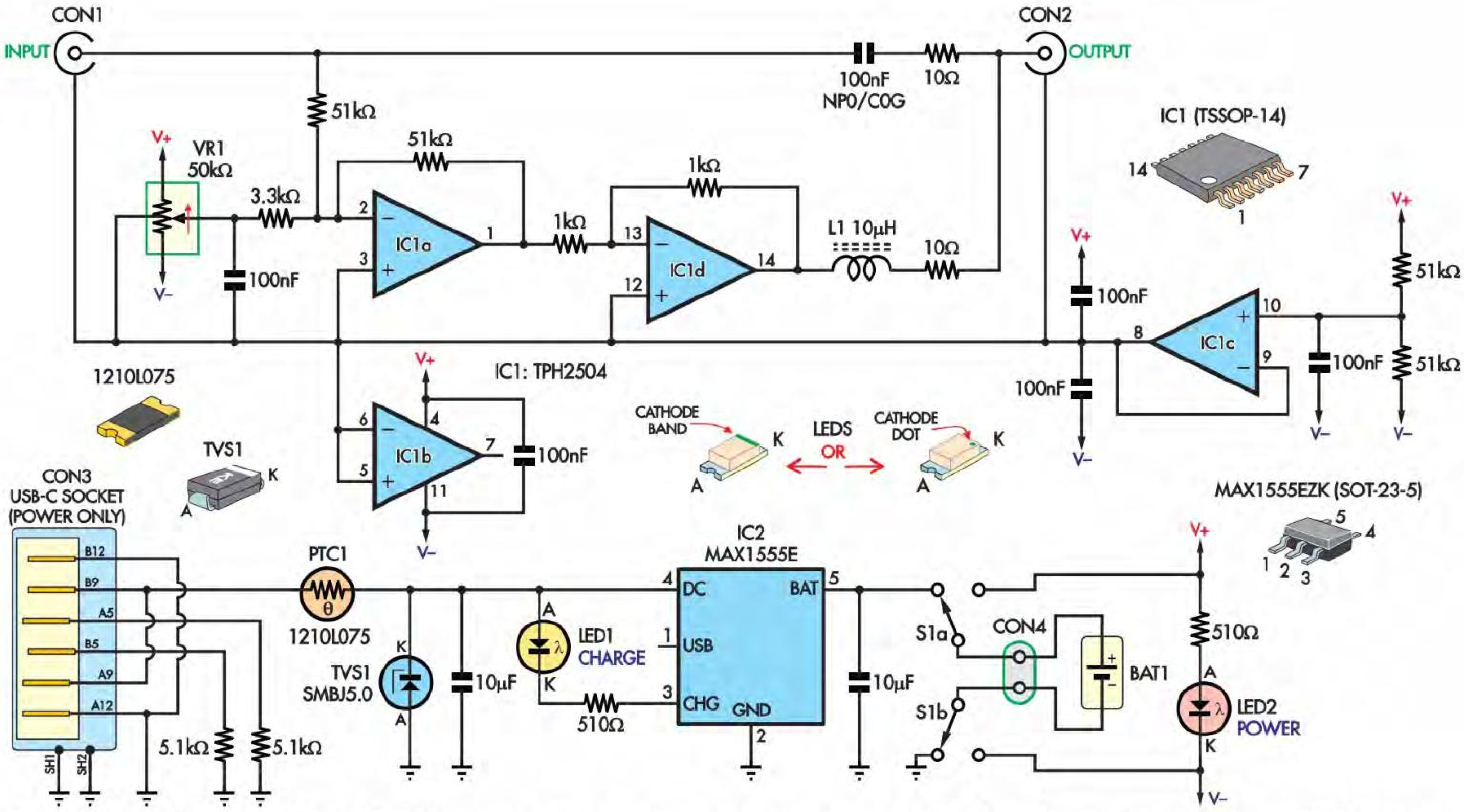

CON1
INPUT
CON2
OUTPUT
100nF
NP0/C0G
10Ω
51kΩ
VR1
50kΩ
V+
V–
3.3kΩ
100nF
51kΩ
IC1a
1kΩ
1kΩ
IC1d
L1 10μH
10Ω
IC1 (TSSOP-14)
14
7
1
1210L075
IC1: TPH2504
IC1b
100nF
IC1c
100nF
100nF
100nF
51kΩ
51kΩ
TVS1
K
A
CATHODE BAND
CATHODE DOT
LEDS
OR
CON3
USB-C SOCKET
(POWER ONLY)
B12
B9
A5
B5
A9
A12
SH1
SH2
5.1kΩ
5.1kΩ
PTC1
1210L075
TVS1
SMBJ5.0
10μF
LED1
CHARGE
510Ω
IC2
MAX1555E
DC
USB
CHG
GND
BAT
10μF
S1a
S1b
CON4
BAT1
510Ω
LED2
POWER
MAX1555EZK (SOT-23-5)

Fig.3: the circuitry is fairly straightforward, using op amps to generate an adjustable offset voltage that's applied to the low-frequency signal path. Potentiometer VR1 is a little unusual in that it has a centre tap and detent, to ensure that its wiper is at signal ground when centred.

since the whole circuit is designed to add an offset. As long as it is no more than a few millivolts, the trace should be on the screen with the pot centred.

We also don't have to worry too much about the op amp's input common-mode range because we are using inverting amplifiers, which have their input voltages fixed at zero, and we have split supplies.

We do need to use op amps that can operate at low supply voltages, and we need a reasonable output capability, since IC1d is driving a 60Ω load, and we'd like to swing as close to the ±1.8V rails as possible. We need the same drive capability for IC1c, as it is driving the other end of the same load.

The most important op amp selection criteria is bandwidth, or more specifically, phase shift; a requirement not necessarily obvious given that the crossover frequency is only 159kHz. You could be forgiven for assuming that an op amp with a bandwidth of a few MHz would be fine in this application.

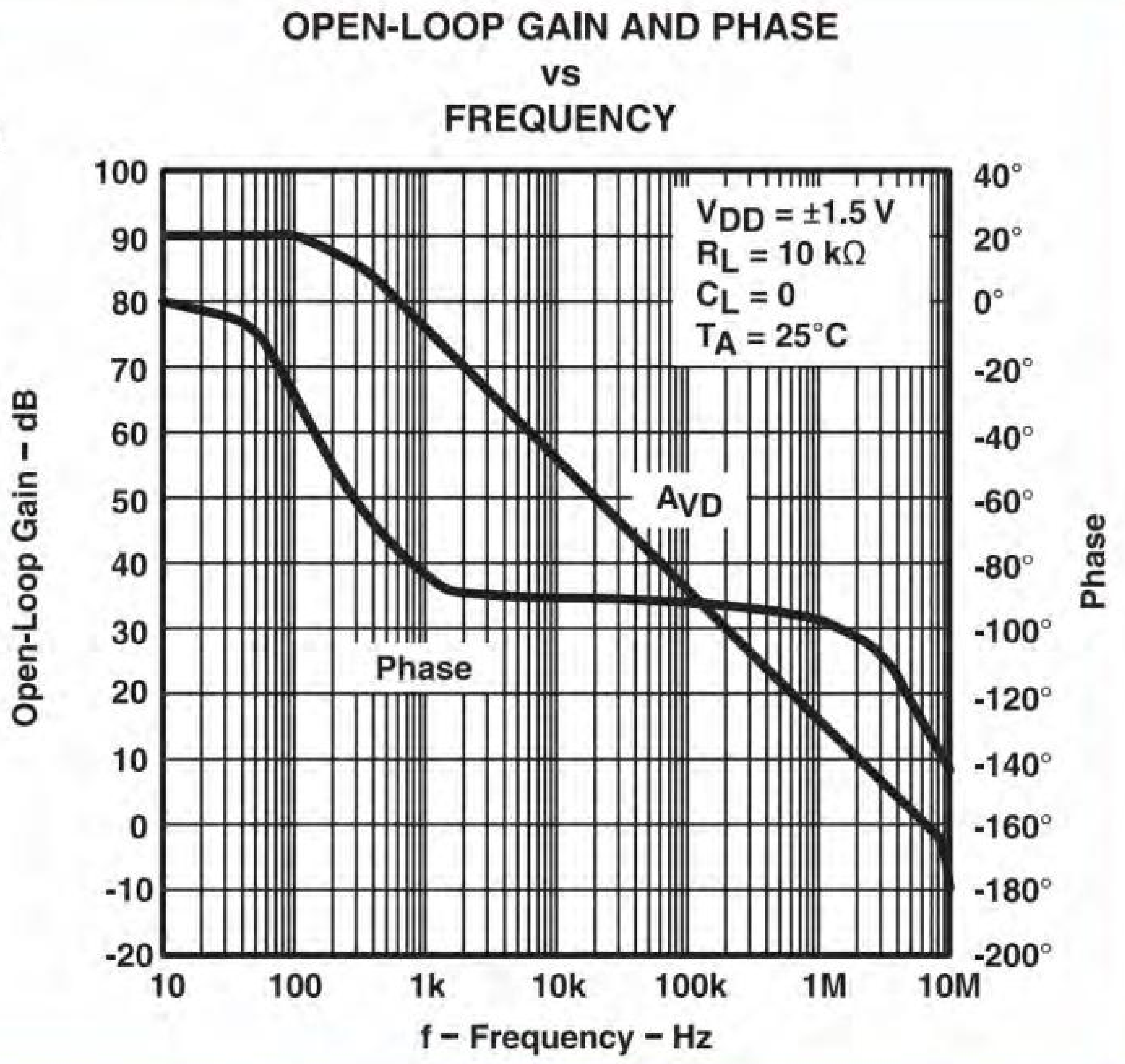
Fig.4 shows the open loop gain and phase of one candidate, the TLV2460 family. These look promising at first, with a rail-to-rail output swing, ±80mA output drive, ±2mV offset voltage and a bandwidth of 6.4MHz. However, close examination of the phase plot reveals a problem.

Most op amps have internal dominant pole compensation that rolls off the open-loop gain response at -20dB/decade, as shown here. It also means the phase shift through the op amp is around -90° over much of its bandwidth. This roll-off is necessary for the stability of the op amp. If the phase shift were to reach -180° before the gain dropped below unity (0dB), the op amp would oscillate.

You can see from the plot that the phase shift through the op amp starts to drop from -90° at around 300kHz, and is down to -100° around 1MHz.

This will be a problem for us, since any deviation from -90° will cause a phase shift in our closed-loop response. If there is an appreciable phase shift through the low-frequency path relative to the high-frequency path, the two signals will add destructively, and we will see a dip in the overall frequency response near the crossover frequency when both signals are contributing to the total.

Fig.4: the open-loop gain and phase plot for the TLV2460, from its data sheet, shows that the phase begins to deviate from -90° at around 300kHz, well below its gain bandwidth (GBW) figure of 6MHz.



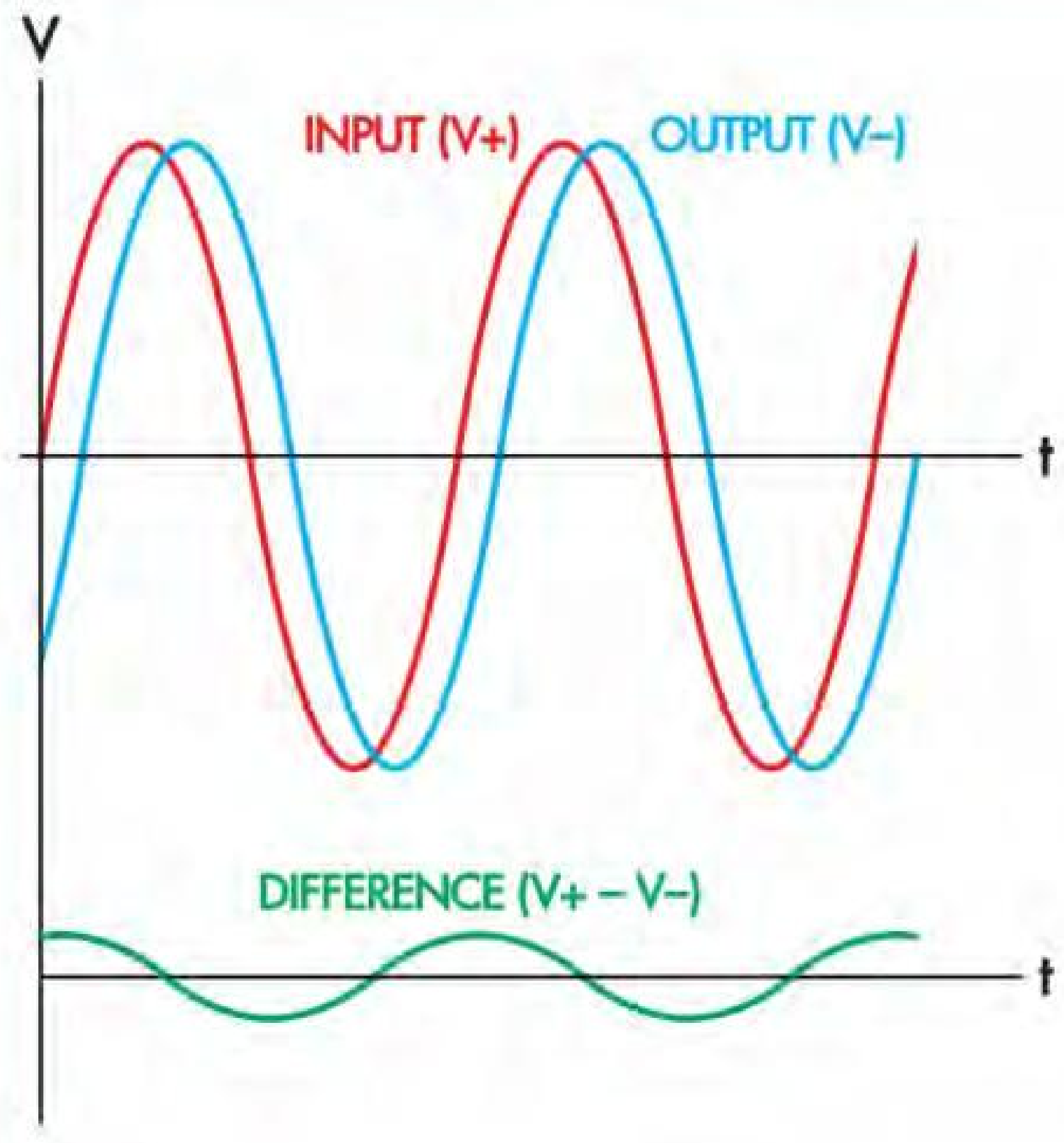
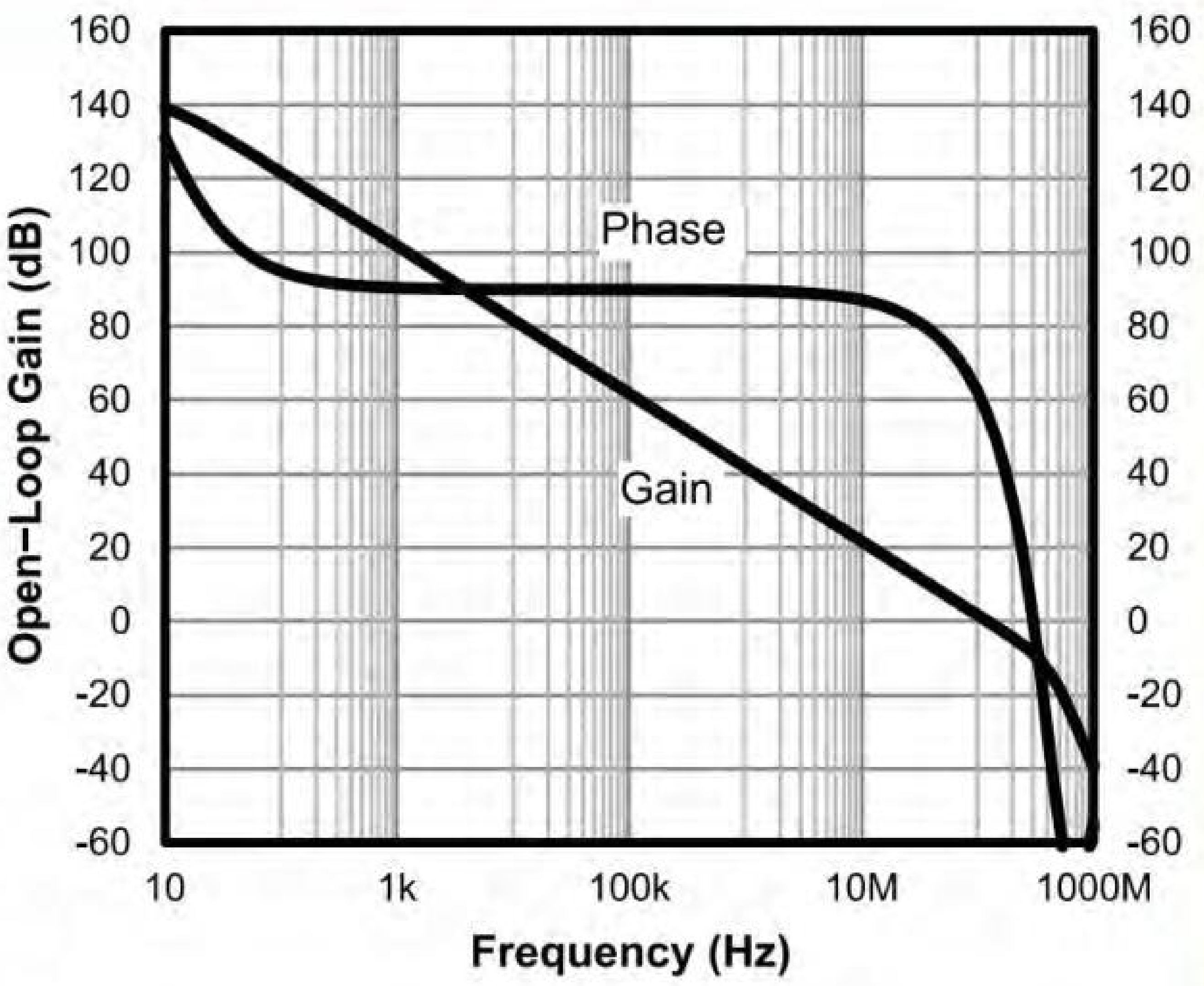


Fig.5: a -90° open-loop phase shift (red trace to blue trace) results in a near-zero closed loop phase shift for a non-inverting amplifier. The phase shifts are exaggerated for clarity in this diagram.

Fig.6: the open loop gain and phase plot for the TPH2504 shows that the phase remains very close to -90° all the way to 10MHz or thereabouts. The horizontal scale of this graph is strange, though.



Op amp phase shift can be a bit hard to wrap your head around. How can an op amp with an open-loop -90° phase shift produce an amplifier with zero closed-loop phase shift (or 180° with an inverting amplifier)?

Hopefully Fig.5 helps explain this. The upper chart shows the input and output voltage waveforms of an op amp configured as a non-inverting buffer.

The red trace is the input voltage applied to the non-inverting input, and the blue trace is the output voltage, which is also applied to the inverting input via the feedback. I have shown an exaggerated phase shift between them to make the point.

The green trace shows the difference between these waveforms. This is the voltage between the op amp's two input pins that is amplified to produce the output. In reality, this voltage will be tiny, due to the high open loop gain of the op amp, but it will not be zero.

You can clearly see that the phase shift between this open loop input voltage and the output voltage is close to -90° because of the dominant pole. If this phase shift were to increase (in the negative direction) to -100° like the TLV2461's data suggests, the phase shift between the input voltage and the output voltage would increase to -10°.

The TLV2460 is therefore going to introduce a significant phase error near to the crossover frequency, and we have two of these op amps in series, doubling the problem.

The solution is to choose an op amp with a much higher bandwidth and/or a much more stable open-loop phase response, up to 10MHz at least.

A bit of searching turned up the TPH2504 family. This is an op amp from 3-Peak – a company I had never heard of until this year. They seem to make some op amps with very impressive price/performance ratios. This one has ±2mV input offset, ±100mA drive capability and 120MHz gain-bandwidth (GBW). A quad pack IC of these costs less than £1.50 in small quantities.

Fig.6 shows the open loop gain and phase plot from its data sheet. I have to say that this is one of the dumbest graphs I have seen in a while, because the horizontal scale increase by a factor of 100 every major division instead of by a decade like every other log-frequency graph you have ever seen. Why?

Nevertheless, you can see that the phase shift remains near -90° all the way to 10MHz.

Capacitor and inductor

There is not much else to say about the signal paths. I used a 100V C0G/NP0 ceramic capacitor in the high-frequency path because we want the capacitance to remain constant with temperature and DC bias. Don't substitute another dielectric like X7R here.

In the low-frequency path, I chose an inductor with a reasonably tight ±5% tolerance and a fairly high (40MHz) self-resonance. A typical inductor has a tolerance of ±20%, so ±5% is pretty good without being unnecessarily expensive

Power supply

Unlike the Differential Probe and the Current Probe, the Power Rail Probe cannot use a DC-DC converter to create the power rails. The last thing we want to do is to inject a bunch of extraneous switching noise into a device that only exists to allow us to measure the switching noise of the circuit under test!

Fortunately, as the required signal amplitude is limited to ±1V, it is feasible to use the battery voltage directly, with the signal common derived from the mid-point as described above. This decision has two design implications.

The previous designs used an unprotected LiPo cell and relied on the under-voltage lockout built into the DC-DC converter IC to prevent over-discharge. Not having this feature means choosing a cell with a built-in protection circuit (or adding a separate protection circuit, which would make the overall circuit more complex).

I also chose to use a standard connector to provide a bit more flexibility regarding cell choice. Any cell with the requisite protection board and a JST PH style connector that fits in the case should work.

The second design implication is that we now have separate 'grounds' for the signal circuit (half the cell voltage) and the charging circuit (cell negative). In most cases, the signal common will be connected to mains Earth via the oscilloscope's BNC terminal. It's also possible (likely?) that the USB charging port, and hence the charging common, will be grounded.

Unless we fully isolate the two circuits, there is the potential for a short circuit. The solution is to use a two-pole power switch to ensure the two circuits can never be connected to each other.

The charging circuit is identical to my previous designs. The input is a power-only USB-C connector followed by a resettable fuse and a 5V TVS pro-

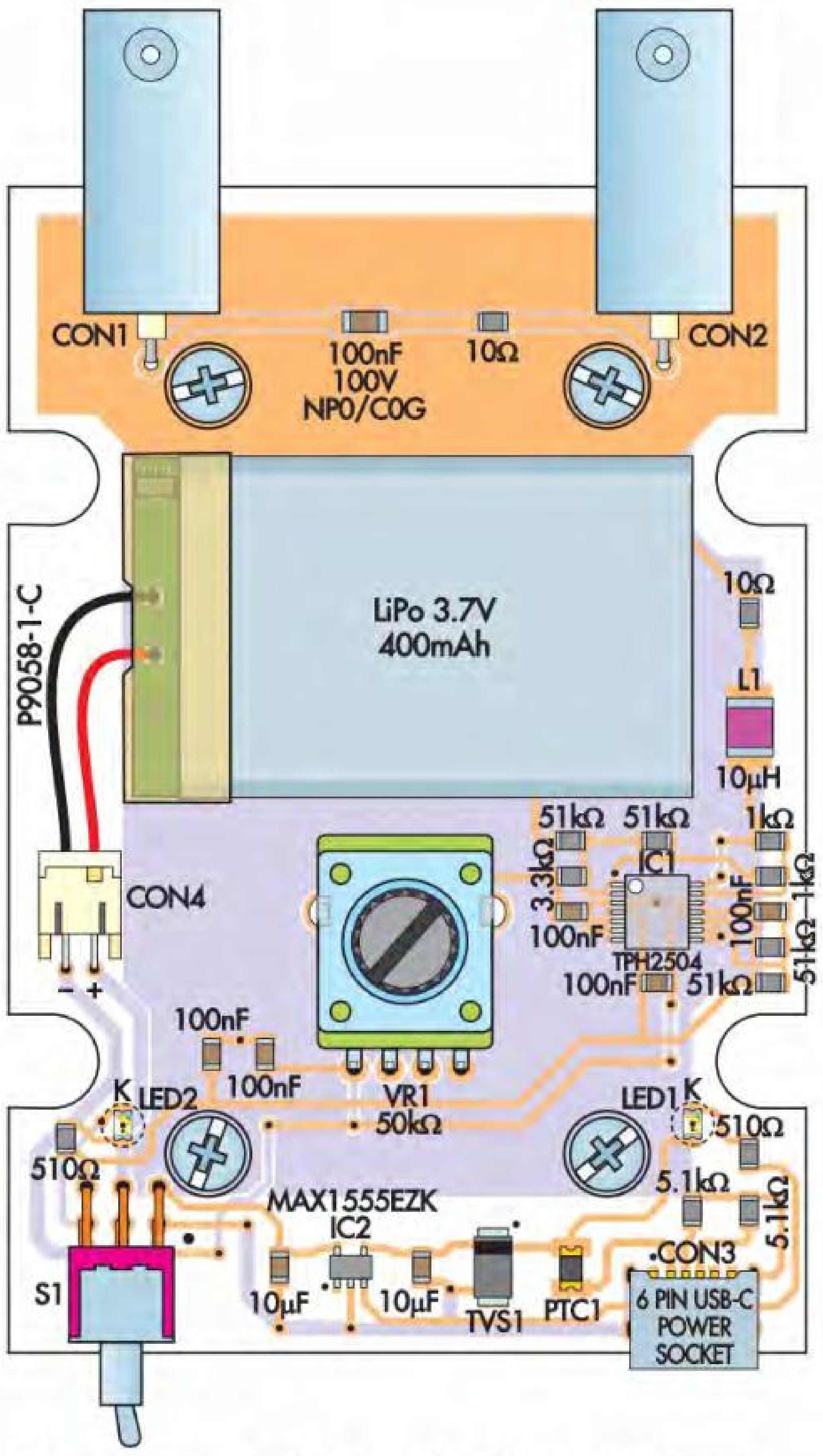


Fig.7: assembly should be quite easy and fast as there are only a few parts. Take care with the orientation of the LEDs, TVS diode and the quad op amp.

Parts List – Power Rail Probe

1 double-sided PCB coded P9058-1-C, 56 × 82mm
1 front panel label, 41 × 60mm
1 Hammond 1593LBK plastic enclosure, 92 × 66mm
2 PCB-mounting right-angle female BNC connectors (CON1, CON2) **[Molex 73100-0105]**
1 USB-C power only socket (CON3) **[Molex 217175-0001]**
1 JST 2.0mm pitch 2-pin right-angle header (CON4) **[JST S2B-PH-K-S]**
1 10µH ±5% 480mA 240mΩ 40MHz SMD inductor, M4532/1812 size (L1) **[Murata LQH43NH100J03L]**
1 0.75A 24V M3226/1210 PTC polyfuse (PTC1) **[Littelfuse 1210L075/24PR]**
1 PCB-mount right-angle DPDT toggle switch with short actuator (S1) **[E-Switch 200MDP1T2B2M6RE]**
1 top-adjust, centre-tapped, centre-detent 50kΩ linear potentiometer (VR1) **[Bourns PTT111-3220A-B503]**
1 400mAh 35 × 25 × 5.2mm LiPo cell with JST PH connector (BAT1) **[eg, Kitronik 2209]**
2 3mm diameter, 0.6in/15.24mm rigid convex light pipes **[Dialight 515-1302-0600F]**
1 knob (to suit VR1)
4 #4 × 6mm panhead self-tapping screws
1 small tube of cyanoacrylate glue (superglue)
1 38 × 25mm foam-cored double-sided tape pad
4 small self-adhesive rubber feet (**optional**)

Semiconductors
1 TPH2504 quad 250MHz RRIO op amp, TSSOP-14 (IC1)
1 MAX1555EZK-T Li-ion battery charger, TSOT-23-5 (IC2)
1 yellow SMD LED, M2012/0805 size (LED1)
1 red SMD LED, M2012/0805 size (LED2)
1 SMBJ5.0CA unidirectional transient voltage suppressor, DO-214AA (TVS1)

Capacitors *(all 50V SMD X7R ceramic, M2012/0805 size, unless noted)*
2 10µF 16V
1 100nF 100V NP0/C0G, M3216/1206 size
5 100nF

Resistors *(all SMD ±1%, M2012/0805 size, unless noted)*
4 51kΩ
2 5.1kΩ
1 3.3kΩ
2 1kΩ
2 510Ω
2 10Ω

tection diode. These are included to protect against a rogue USB-C source applying a voltage higher than 5V to the connector. The two 5.1kΩ resistors signal the USB C power source to supply 5V at up to 3A.

Yellow LED1 illuminates when the LiPo cell is charging and goes out when full charge is reached. The green LED (LED2) indicates that the unit is switched on. The charger, IC2, is configured to provide a 280mA charging current, so it should recharge a 400mAh cell in under two hours.

The overall operating current consumption is 25-50mA depending on the signal level, so the battery life should be 8-16 hours.

Construction

All components mount on a single 56 × 82mm PCB coded P9058-1-C. For once, there are no tiny leadless parts, so assembly requires nothing but a soldering iron and a steady hand. You can commence by fitting the surface-mount parts according to the overlay diagram, Fig.7. Watch the polarity of the LEDs, the TVS diode and the TSSOP quad op amp. The rest don't matter, or are hard to get wrong.

The USB connector has surface-mount pads for the terminals, as well as through-hole mounting pads. The best way to mount this is to first solder it in place via the through-hole pads from the bottom, then turn the board over and solder SMT pads.

Finish the PCB assembly with the battery connector, the BNC terminals, the switch and the pot. That's all there is to it.

Testing

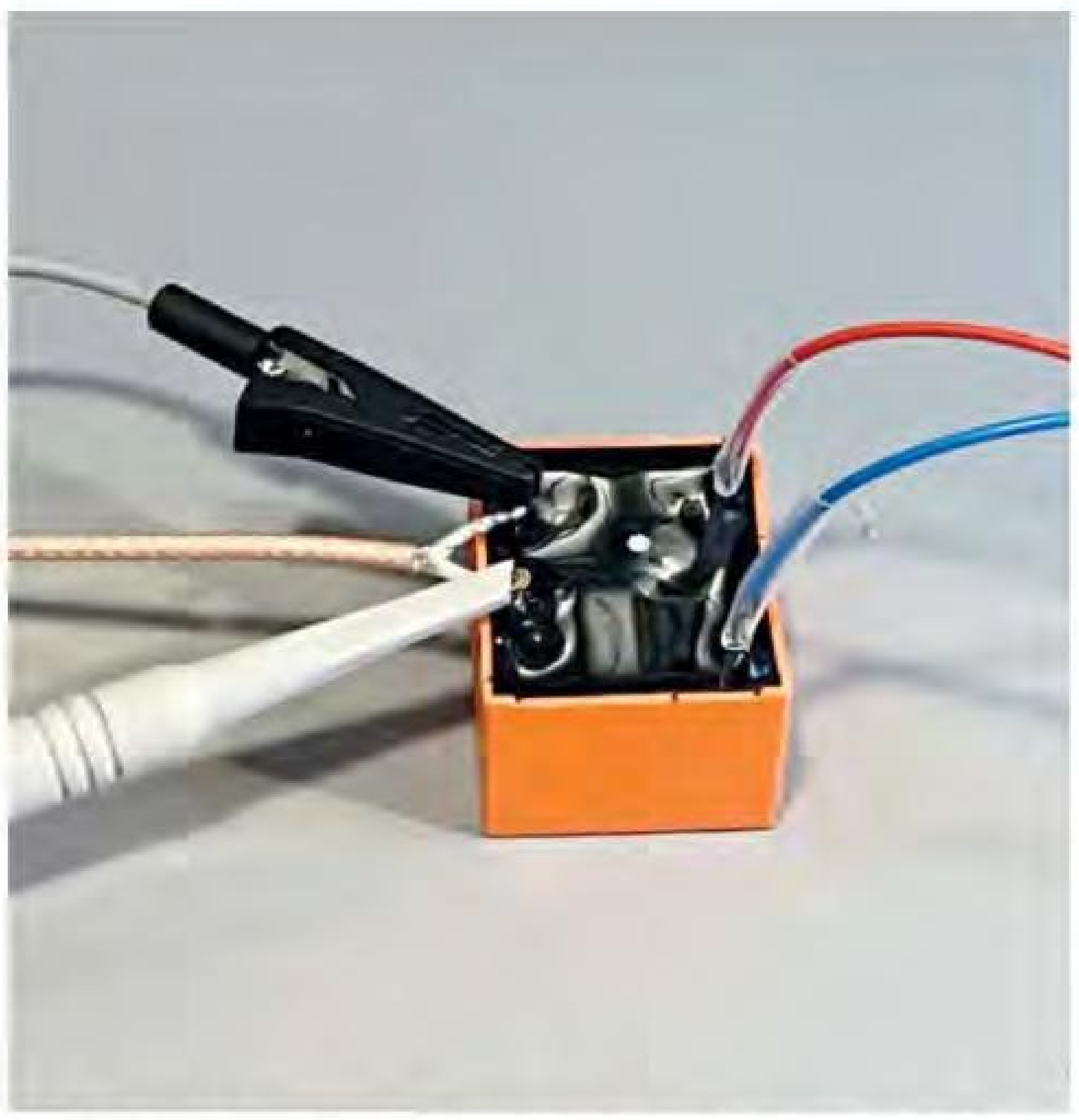
Check your work carefully, then connect the battery or an external supply set to 4.0V. Switch it on and you should see the green LED light. Use a multimeter to measure the power supply voltages with reference to one of the BNC connector shields.

The bottom ends of the two 100nF capacitors just to the left of VR1 are convenient places to probe. You should read around +2V on the left-most capacitor and -2V on the one to its right. Anything between ±1.8V and ±2.2V is fine.

You can check the DC offset with the pot centred by measuring the voltage between the centre pin and shield of the output BNC connector. The voltage should be within ±5mV of zero. You can check the output voltage swing with the same set-up. Simply turn the pot either way until the output saturates. The voltages should be well above ±1.0V, even at the lowest battery voltage.

With a fully charged battery, they will be closer to ±1.3V.

You can check the battery charger is working by switching the unit off and connecting a USB-C power source.



The test set-up used a conventional scope probe and a home-made RG316 probe to measure the output of this AC-DC converter module.

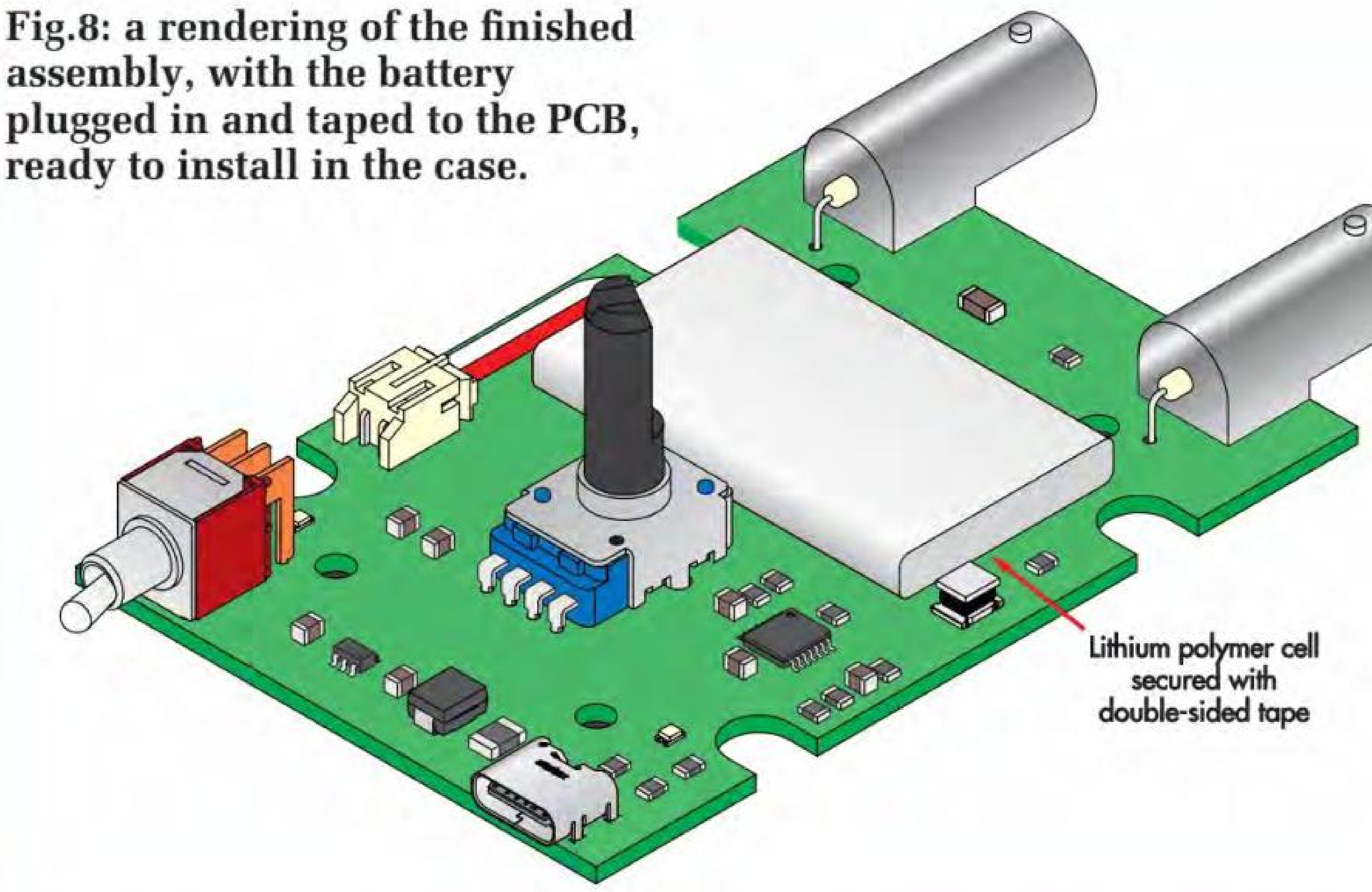


Fig.8: a rendering of the finished assembly, with the battery plugged in and taped to the PCB, ready to install in the case.

Unless the battery is fully charged, the yellow LED should light, and the battery voltage should climb slowly. The LED extinguishes when the battery reaches full charge, at around 4.2V.

Final assembly

You can now fix the battery in place with a small piece of double-sided tape, as per Fig.8, then turn your attention to the case. Mark out and drill the two end plates and the top according to Fig.9.

The aperture for the USB connector is best opened up after drilling by using a sharp craft knife or scalpel to remove the material between holes drilled at each end. I used a few small files to neaten things up.

You can then apply the label to the top surface of the lid. The artwork is available to download (siliconchip.au/Shop/11/2771).

I printed mine full-size on glossy adhesive paper, then laminated that with some transparent self-adhesive vinyl. Cut it to size and fix into the recess in the lid, starting at one end to avoid capturing bubbles.

I opened up the two light-pipe holes by pushing a sharp probe through the label into the holes in the case. The pot shaft opening is large enough to use a blade to remove the label over the aperture.

Install the light pipes from the top of the case, and secure them on the underside with a drop of superglue.

Thread the end panels onto the PCB assembly and lower it into the bottom of the case, making sure the end panels go into the slots provided for them. The board is held down by four 6mm-long #4 self-tapping screws. Pop the top case on and secure with the screws provided.

I added four small self-adhesive rubber feet on to the bottom of the case. Fit the knob and you are finished.

Using it

The Power Rail Probe is dead easy to use. Connect the output to your oscilloscope with a 50Ω BNC cable and set the input to 50Ω termination. Set the vertical scale to a few hundred millivolts initially. Connect the Power Rail Probe's input to your circuit and switch it on.

With your circuit under test powered up, you should be able to adjust the offset pot to get the scope trace very close to zero.

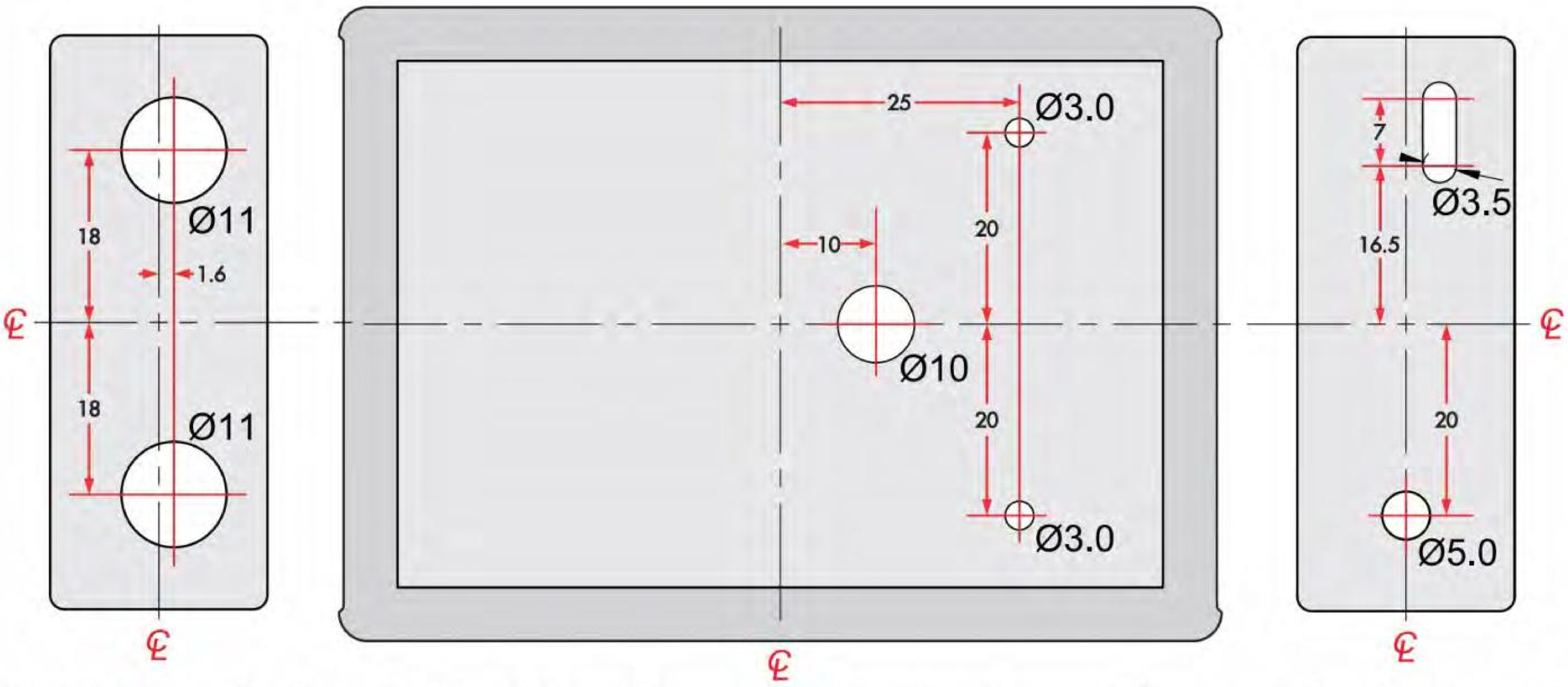
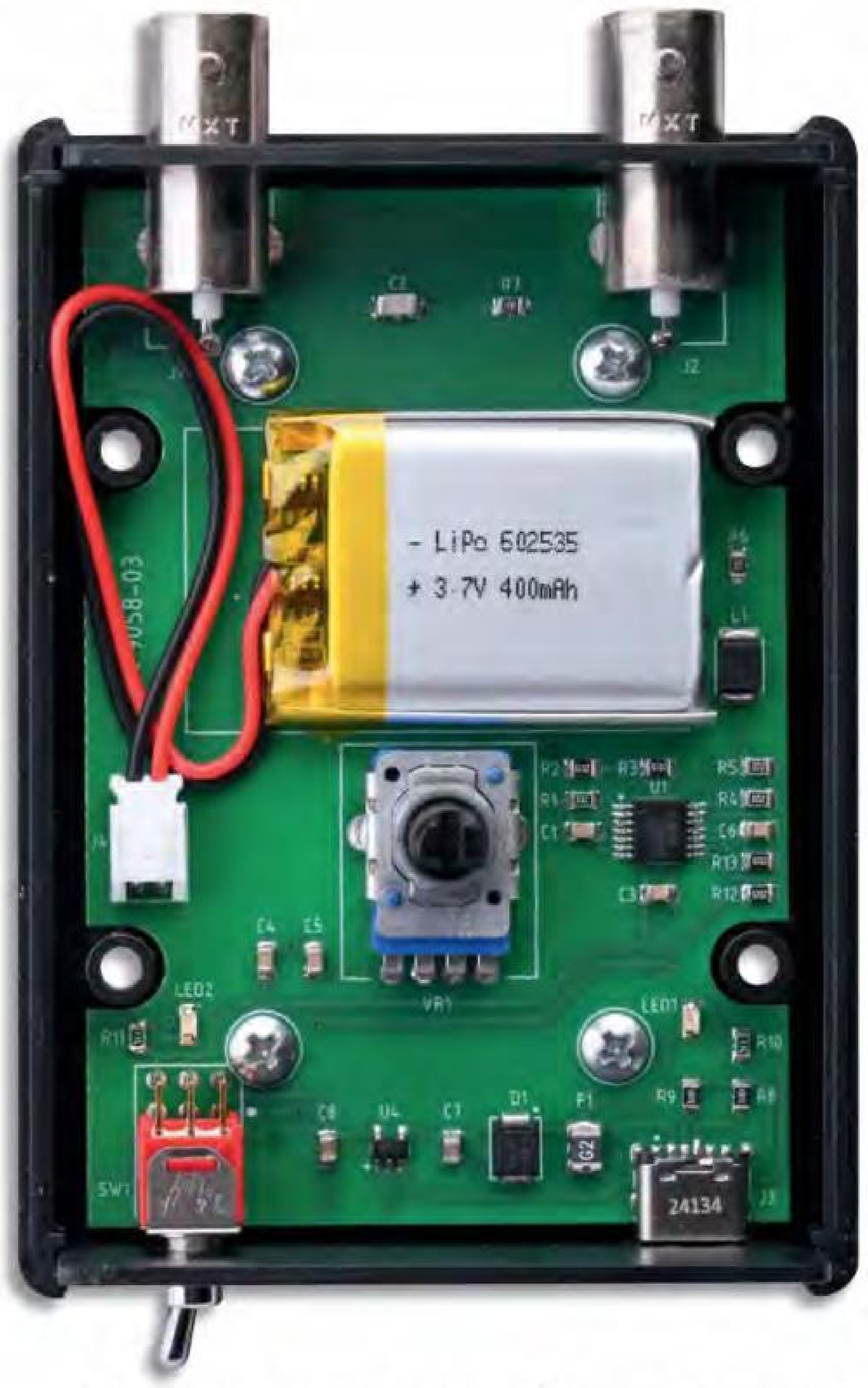
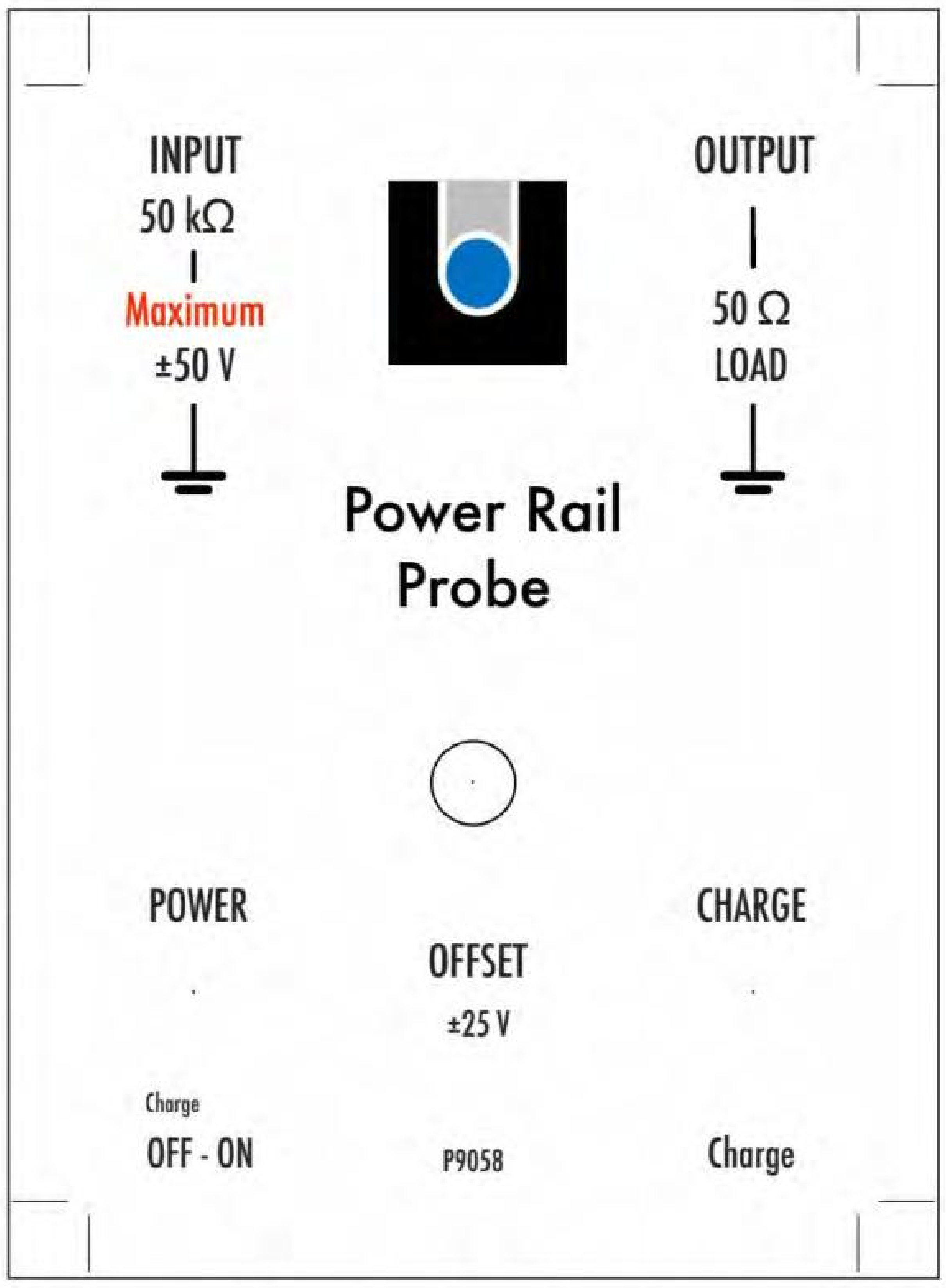


Fig.9: drill the top and the flat end-plates of the enclosure according to this diagram. The contoured end-plates supplied with the case are not used.



The finished Power Rail Probe, mounted in its 99 × 66mm plastic enclosure.



The front panel label for the Power Rail Probe. Note the very small dots below the POWER and CHARGE labels. These are for the SMD LEDs, which shine through the panel via 3mm diameter light pipes. Punch 3mm holes centred on those dots after the label is affixed to the case.

You can now zoom down the vertical scale appropriately, tweaking the offset pot slightly if necessary to keep the trace centred on the screen. Just remember the output on screen is attenuated by about 1.2 times (the attenuation will be within the range of 1.1-1.3 times).

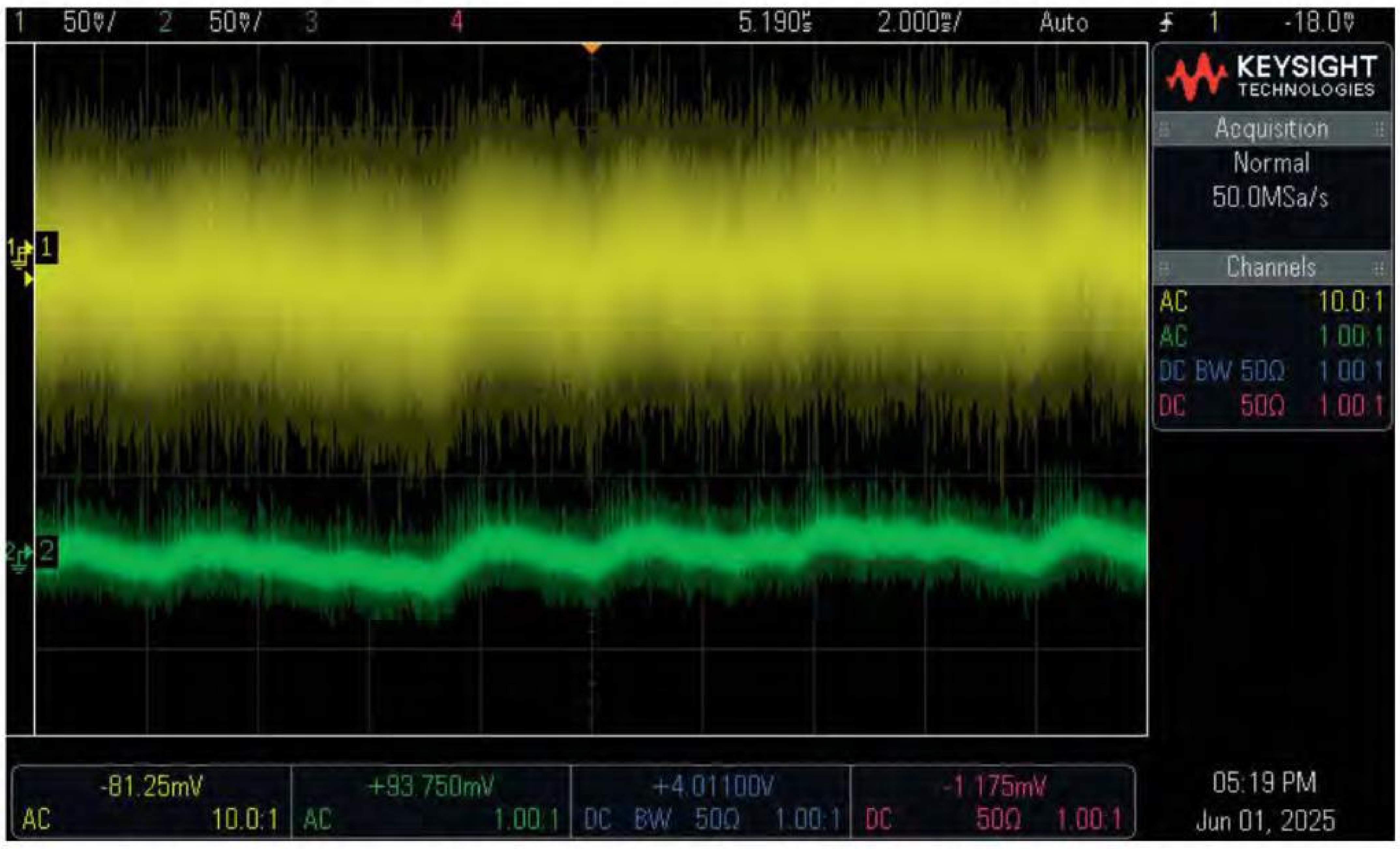
The connection you make between the power rail probe and your device under test will be the single most important factor in the measurement's usefulness.

Probing any high-frequency signal can be difficult, especially when you are working with switching power supplies. They tend to be environments rich in radiated and conducted interference that can easily upset your measurements.

I set up a small experiment to demonstrate this, using a Zettler modular AC-to-DC converter rated at 15V and 5W. This is the one used in the Variable Speed Drive for Induction Motors project (from the December 2025 & January 2026 issues).

I measured the unloaded output voltage of this switch-mode module with a conventional passive oscilloscope probe, with a 150mm ground clip and with a custom 'probe' made up of a short length of RG316 coax with a BNC connector fitted to one end. The photo at left shows the test setup.

The scope capture (Screen 1) tells the story. The scope probe's ground loop acts as a very effective antenna to pick up all sorts of switching hash radiating from the converter module. As a result, the underlying ripple is more-or-less invisible below the noise in the yellow trace from the probe. The home-made probe (green trace) has a much smaller loop area and picks up proportionately less noise.



Scope 1: the results from the test shown at upper left. The waveform measured by the standard probe (yellow trace) is completely buried in switching noise, while the green waveform from the Power Rail Probe is much more informative.

The faint vertical spikes you can see on the green trace are real signal artefacts caused by the very high voltage rates-of-change in the primary switch being capacitively coupled to the output. Their irregular spacing shows that the converter is operating in burst mode due to the very light load.

Using coax probes like this is not something I invented. Commercial power rail probes come with similar unterminated cables for this purpose. However, there is a much cheaper alternative. I buy 1m RG316 BNC-to-BNC cables from AliExpress and cut them in half to yield two test probes. You can reuse them many times, but they eventually get too short and have to be discarded.

At the time of writing, three such cables cost less than £12.50 delivered. That's way less than the cost of buying the cable and connectors to making them myself.

Conclusion

A Power Rail Probe is far from the most essential piece of test equipment you will ever own. However, if you are looking at the dynamic response of converters that take place over tens of milliseconds, at voltage levels where you may run out of DC offset in your oscilloscope, there may be no alternative. Building this Probe is likely to be the most cost effective way to get that capability. PE

By Max the Magnificent

WEIRD & WONDERFUL ARDUINO PROJECTS

Part 16: etching sketches and snake shenanigans

Hi there, I hope you are feeling anticipendulous and exhilarocious regarding the cunning code we are about to create concerning our retrodocious games console. Why yes, I have indeed delighted in both of the *Wicked* movies. How did you guess?

As you may recall, my retro gaming console features a laser-cut front panel mounted in a 3D-printed enclosure. From the start, I noted that readers without access to these tools could construct their own consoles by simply attaching the various parts, such as the 7-segment display modules, 14 × 10 tricolour light-emitting diode (LED) array, and push-button switches, to a sheet of timber.

I just heard from a reader we'll call Ian (because that's his name). Ian was keen to tell me about the current state of his console. He enclosed some snapshots of his handiwork (Photos 1 and 2).

In his message, Ian commented:

As you can see, I don't have access to a CNC machine or 3D printer, so I had to resort to my other passion: woodworking. My console is formed from MDF mounted on a couple of pieces of wood, but it seems to be working out OK so far.

Last month, I decided to deviate slightly from your design and build my console with an onboard Nano instead of separate cartridges. I have an internal extension cable, so I can plug a USB cable into the console to download games as required.

I, for one, think Ian's ingenious implementation has turned out brilliantly. This is just what I was envisaging. Hopefully, this will prompt other readers who have been sitting on the "pity me because I don't have a CNC" fence to leap into action and join in the fun.

Before we proceed, we should perhaps note that, in Ian's realisation, the four black north (N), south (S), east (E), and west (W) direction control buttons are located on the lower left-hand side of his console. (These buttons are white in my implementation.) Meanwhile, the four red (R), yellow (Y), green (G), and blue (B) action control buttons are located on the lower right-hand side of Ian's panel.

This was the way I'd initially planned to do things myself, and Ian was simply following my earlier instructions. However, due to an unexpected turn of events (I messed things up), I ended up with my action buttons on the left and my direction buttons on the right. Oh well, what can you do, eh?

Ooh, shiny!

Continuing our gaming console saga, in our previous column, we created a "framework program" that we can think of as the scaffolding that does all the heavy lifting behind the scenes.

To ensure we're all tap-dancing to the same skirl of the bagpipes, you can revisit this program in the file named **CB-apr26-code-01.txt**. As usual, all files mentioned in this column are available as part of the April 2026 download package from the PE website at **https://pemag.au/link/acau**

We subsequently used this framework as the basis for a simple program that let us use our RYGB action control buttons to modify the colour of a flashing pixel on our 14 × 10 array. Next, we augmented this program to use the four NSEW direction control buttons to move the flashing 'pixel' around the array.

Finally, we taught the console to keep track of how many times we press any of the action control buttons and display this value in the three leftmost 7-segment digits located at the top of the console. Similarly, for the number of times we tap any of the direction control buttons, which we display in the three rightmost 7-segment digits.

If your controls are arranged à la Ian, you might decide to swap which 7-segment digits display the various values. You can peruse and ponder the way we left this program in the file named **CB-apr26-code-02.txt**.

Electronic Etch A Sketch

The Etch A Sketch was invented in 1959 by a French electrician called André Cassagnes. By the late 1960s and early 1970s, this toy was a staple Christmas stocking-stuffer in the UK. I think it's fair to say that almost every kid I knew had one.

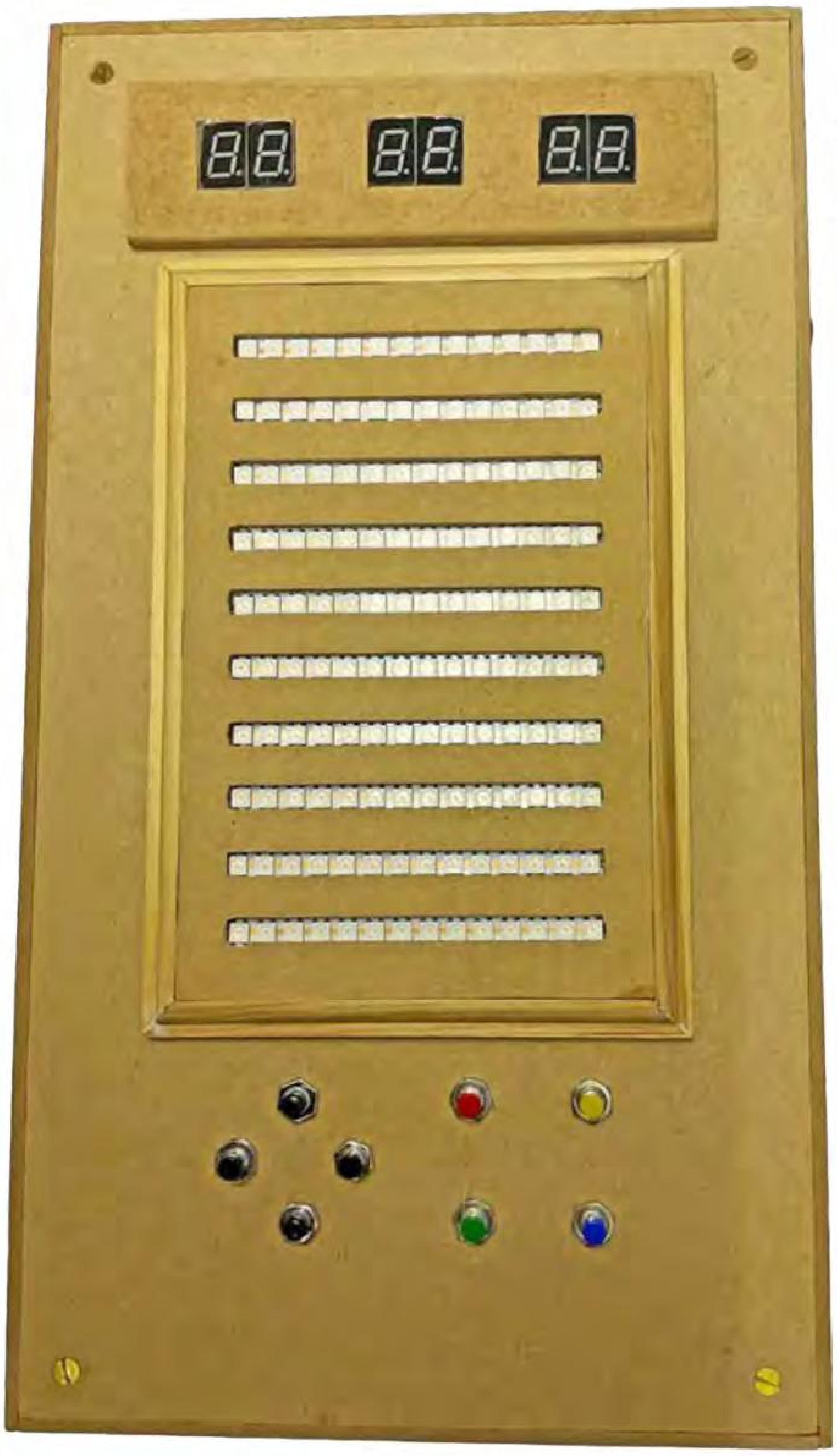
The Etch A Sketch is one of those rare toys that never really went away. Over the decades, it smoothly transitioned into 'timeless toy' territory, eventually repositioning itself as a retro classic that's still manufactured and sold to this day.

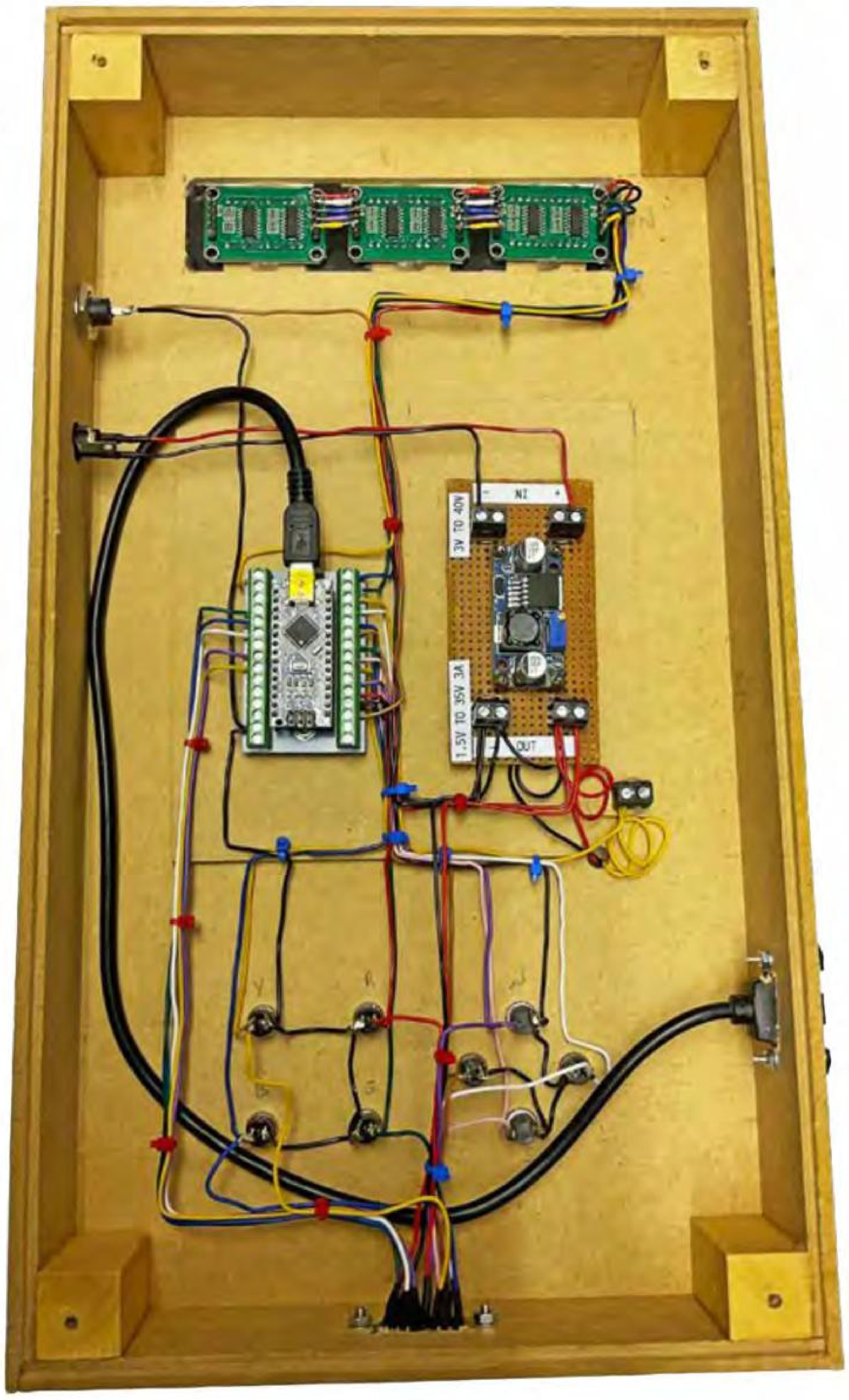
I know what you're thinking. Did he fire six shots or only five? Sorry, wrong script. You are thinking to yourself: "Why don't we use our retro games console to implement an electronic Etch A Sketch?" I was just thinking the same thing. "Great minds think alike", as they say. Of course, they also say, "Fools seldom differ", but that certainly doesn't apply to us!

So, what will it take to convert our existing program into the first pass at a rudimentary electronic Etch A Sketch? Surprisingly little. In fact, all we need to do is tweak a single line of code. Consider the **UpdateLocation()** function in the current version of our program, as illustrated in Listing 2(a).

Remember, we're using the convention that the listing number (2 in this case) corresponds to the numerical part of the matching code file name ("02" here).

This function is called whenever





Photos 1 and 2: Ian's MDF-based console.

we press one of our NSEW direction control buttons. The first thing we do on line 461 is set the current pixel to the background colour (currently **BLACK**). On lines 462 and 463, we update our X-Y coordinates to reflect their new values.

Then, on line 464, we call our **UpdateFlash()** function to force the pixel at our new X-Y location to start flashing with the current foreground colour.

Can you spot which line we need to change? You're right, it's line 461. Instead of setting the original pixel to the background colour, we set it to the foreground colour. I've just made this change (in the file named **CB-apr26-code-03.txt**), as illustrated in Listing 3(a). Now, as we press our direction control buttons, we leave a trail of glowing pixels behind us.

Any colour you want

The famous phrase, "You can have any colour you want, as long as it's black", is attributed to Henry Ford regarding his revolutionary Model T cars. This reflected his focus on mass production with a single, fast-drying black paint to maximise efficiency on the assembly line, thereby making cars affordable and widely available, although the exact wording (and timing) of the quote is debated.

Paradoxically, the only new colour we want for our Etch A Sketch program is black, but there's no way for us to achieve it… or is there?

The problem with the updated version of our code is that we cannot prevent ourselves from leaving a trail of illuminated pixels as we manoeuvre our flashing cursor around the array. This is unfortunate because sometimes we may wish to move the cursor without marking the display as we go.

My knee-jerk reaction when I first considered this conundrum was to think, "I wish I had decided to include an additional black (K) action button located in the centre of my existing RYGB action buttons."

```
459  void UpdateLocation(int8_t deltaX, int8_t deltaY)
460  {
461    DrawPixel(X, Y, BackGndColor);
462    X = (X + NUM_COLS + deltaX) % NUM_COLS;
463    Y = (Y + NUM_ROWS + deltaY) % NUM_ROWS;
464    UpdateFlash(FORCED);
465
466    DirectionCtr = UpdateCounter(DirectionCtr, 1, MIN_COUNT, MAX_COUNT);
467    NumberToBuffer(DirectionCtr, 3, 0);
468    BufferToDisplay();
469  }
```

Listing 2(a): our original **UpdateLocation()** function.

```
459  void UpdateLocation(int8_t deltaX, int8_t deltaY)
460  {
461    DrawPixel(X, Y, ForeGndColor);
462    X = (X + NUM_COLS + deltaX) % NUM_COLS;
463    Y = (Y + NUM_ROWS + deltaY) % NUM_ROWS;
464    UpdateFlash(FORCED);
465
466    DirectionCtr = UpdateCounter(DirectionCtr, 1, MIN_COUNT, MAX_COUNT);
467    NumberToBuffer(DirectionCtr, 3, 0);
468    BufferToDisplay();
469  }
```

Listing 3(a): our modified **UpdateLocation()** function.

Of course, doing so would have introduced its own problems, not least the lack of available digital input/output (I/O) pins. The Arduino Uno has 14 such pins. Pins 0 and 1 are off the table because they are used to communicate with the host PC, leaving us with 12.

We are using one pin to drive our 14 × 10 array and three pins to drive the chain of 74HC595 8-bit serial-in, parallel-out (SIPO) shift registers that live in our 7-segment display modules, leaving us with eight pins to access our eight existing switches. This means that, using our current scheme, there would be no pins remaining to connect to any additional buttons.

Shift over!

One solution would be to add two or more 74HC597 8-bit parallel-in, serial-out (PISO) shift registers. As we see from their data sheet (**www.ti.com/lit/ds/symlink/cd54hc597.pdf**), these behave in a complementary manner to their 595 cousins. We can load the values from eight switches into a 597 simultaneously (in parallel), then clock these values out serially (one after the other) on a single data line.

As with the 595s, it's possible to daisy-chain multiple 597s together. This means we can effectively add as many switches as we wish while using only three of our Arduino's pins. For example, with six 597s, we could support up to 48 switches (see Fig.1).

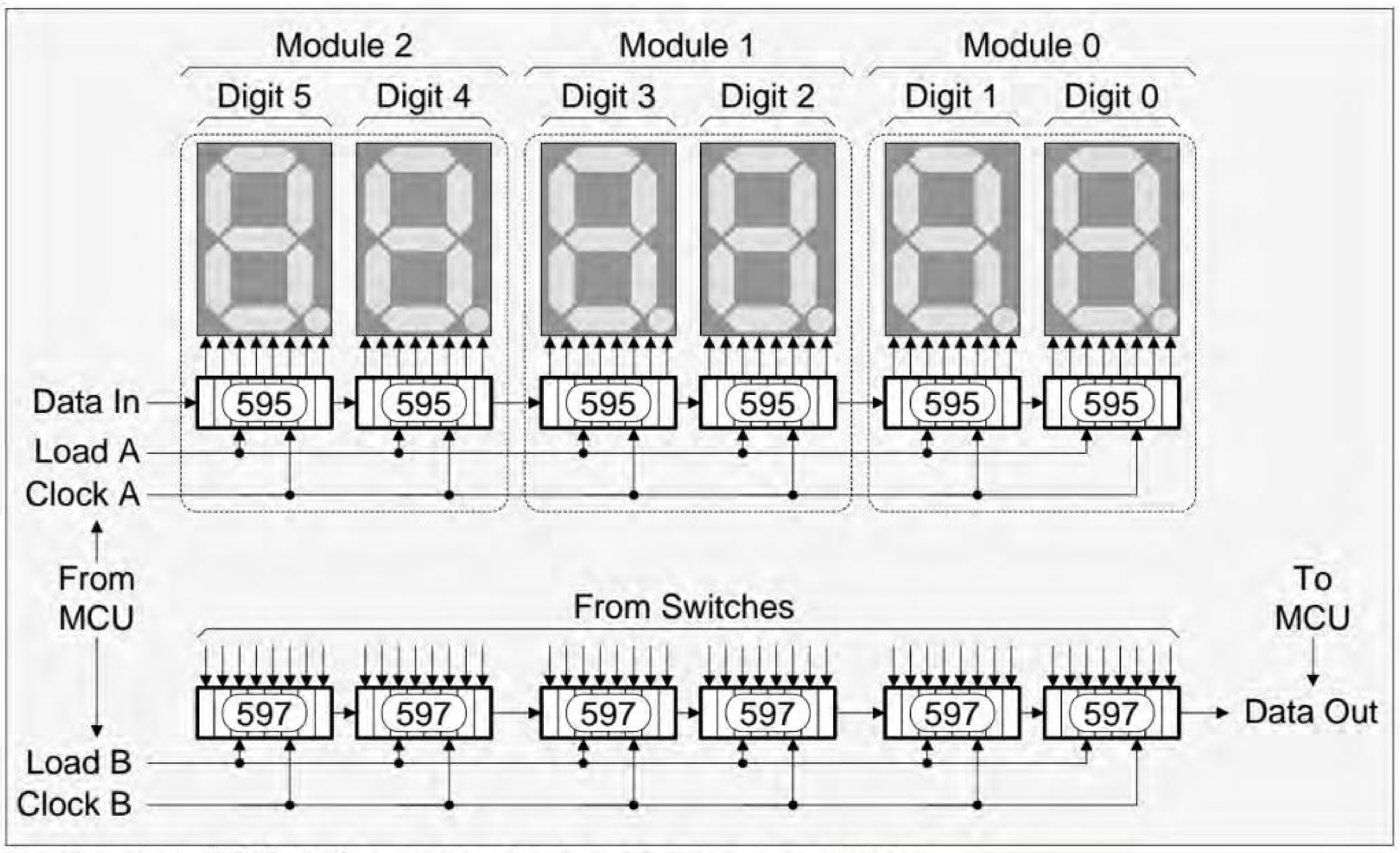


Fig.1: using PISO shift registers to support more switches.

This is tempting for several reasons, but I've made the decision to make do with the eight switches we have. The reason for this will become clear shortly.

Teaching old buttons new tricks

In our current code, whenever we press one of our RYGB action buttons, our switch-handling routine calls the UpdateColor() function, passing it the colour associated with that button. The existing version of this function is shown in Listing 3(b).

```
446 void UpdateColor(uint32_t color)
447 {
448   if (ForeGndColor != color)
449   {
450     ForeGndColor = color;
451     UpdateFlash(FORCED);
452   }
453
454   ActionCtr = UpdateCounter(ActionCtr, 1, MIN_COUNT, MAX_COUNT);
455   NumberToBuffer(ActionCtr, 3, 3);
456   BufferToDisplay();
457 }
```

Listing 3(b): our original UpdateColor() function.

On line 448, we test whether the current foreground colour differs from the colour we've just passed in. If so, on line 450, we set the foreground colour to be the new colour. Then on line 451, we call our UpdateFlash() function to force the pixel to start flashing with the new foreground colour (ie, the one associated with the button we just pressed).

Observe that we don't mess with the background colour because, thus far, we've assumed that it's always set to BLACK. Well, that's about to change. I've just modified this function, as illustrated in Listing 4(a) from the file named **CB-apr26-code-04.txt**.

```
446 void UpdateColor(uint32_t color)
447 {
448   if (ForeGndColor == color)
449   {
450     ForeGndColor = BLACK;
451     BackGndColor = WHITE;
452   }
453   else
454   {
455     ForeGndColor = color;
456     BackGndColor = BLACK;
457   }
458   UpdateFlash(FORCED);
459
460   ActionCtr = UpdateCounter(ActionCtr, 1, MIN_COUNT, MAX_COUNT);
461   NumberToBuffer(ActionCtr, 3, 3);
462   BufferToDisplay();
463 }
```

Listing 4(a): our modified UpdateColor() function.

As you can see, I've modified the test on line 448. In this new version, we check whether the current foreground colour is the same as the colour we've just passed in. If so (for example, the foreground colour is RED and we just pressed the R button), then we set the foreground colour to BLACK and the background colour to WHITE.

Otherwise, we set the foreground colour to the colour of the button we just pressed, and we reset the background colour to BLACK.

In effect, we've made our four action buttons context-sensitive. Pressing any of these buttons once selects that button's colour; pressing it again toggles us into erase mode; and pressing any action button while we're in erase mode returns us to that button's colour.

One more thing

As a final tweak, when we created our original flashing pixel program that evolved into this month's Etch A Sketch program, we could only display the four RYGB colours. This meant that we were obliged to select one of these colours as our default start colour. We picked green because that's my favourite.

However, now that we can display black as a colour, it makes sense to set our default foreground and background colours to BLACK and WHITE, respectively (see the file named **CB-apr26-code-05.txt**).

This means that we can move our

flashing white pixel around the array without leaving a coloured trail until we reach the location from which we wish to commence our drawing. While this is only a minor modification in the scheme of things, I find it both aesthetically and logically pleasing.

Although our Etch A Sketch implementation is simple, it's surprisingly enjoyable to play with. Kids of all ages will love it (I know I do)!

In fact, I was just thinking that we could all have a little competition to see who can create the best image using the current incarnation of our code. If you wish to email your entry to me, I may include some of the more interesting offerings in a future column.

Thanks for the memory

Before we proceed, it's worth noting that our original flashing pixel program consumed 4430 bytes (~13%) of flash and 126 bytes (~6%) of RAM. How much more memory do you think we've consumed by transforming it into our Etch A Sketch program?

The answer is surprisingly little. Due to the way we architected things, we required only small modifications to a couple of core functions. As a result, we added only 78 more bytes of flash consumption and 4 more bytes of RAM, resulting in a usage of 4508 bytes (still ~13%) of flash and 130 bytes (still ~6%) of RAM.

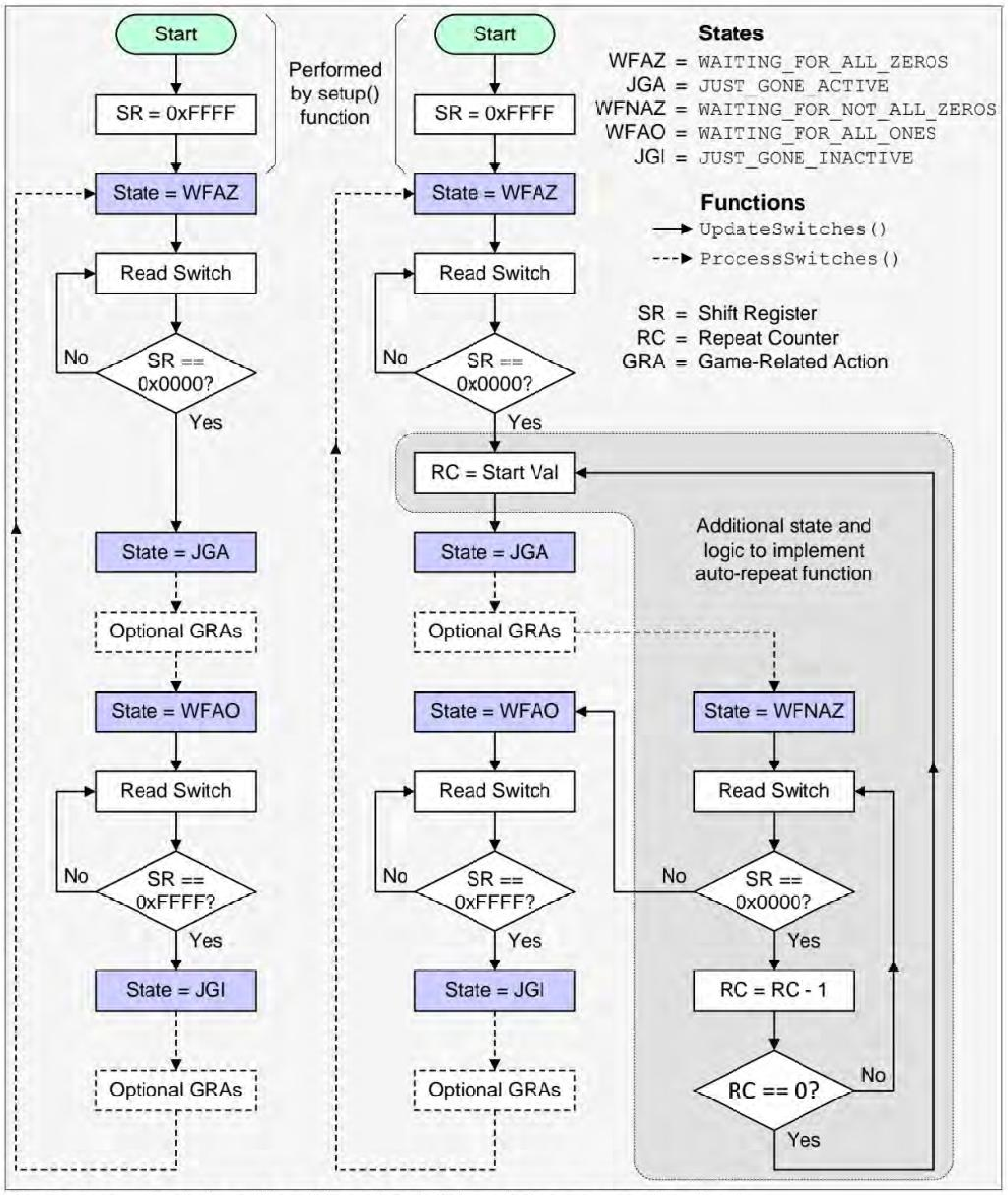


Fig.2: our switch-handling 'flowchart' (all switches are active-low).

Arrgggh (sincerely)

I have a confession to make: I've made a mistake. I know, I know... I find this as hard to write as you do to read, but even awesome Albert (my uncle, not the other one) had his off days.

In the past, we've noted that when we instantiate our array of NeoPixels, the system reserves 24 bits of memory for each array element. It uses these to store the three 8-bit RGB values associated with each pixel. In our case, since we are working with a 14 × 10 array, this amounts to 3 × 140 = 420 bytes of memory.

What we've neglected to consider is where these memory locations live. While they reside in RAM, the problem is that, because the NeoPixel library allocates this memory dynamically (don't ask), it is invisible to the Arduino's integrated development environment (IDE), which is why we've failed to account for it.

The bottom line is that although the IDE reports we're using only 130 bytes of RAM, we're actually using 130 + 420 = 550, which is ~27% of the 2048 bytes (2KiB) available to us. This isn't as innocuous as we'd thought, but it's still not worrisome enough to warrant us wearing our frowny faces.

Idly pondering

For reasons I no longer recall, I was idly pondering the possibility of us adding 'auto-repeat' functionality to our buttons. For example, rather than pressing the east (E) button multiple times, I was wondering if it would be advantageous to allow pressing and holding the button to automatically repeat the action.

Before we plunge headfirst into the mire, it's worth taking a few moments to remind ourselves how we are currently debouncing and processing our switches. Each switch is associated with a simple state machine, as illustrated in Fig.2(a).

Fig.2 shows state machines presented as pseudo-flowcharts rather than traditional state diagrams or program execution flowcharts (that's just the way they fell out when I was drawing them). This means we can't directly translate these diagrams into code; they're just a way of visualising what's going on.

Each major operating state is shown as a purple rectangle. White diamond shapes represent transition decisions, while white rectangles reflect actions to be executed.

In our current code, as shown in Fig.2(a), we decided that our state machine should have four states: **WAITING_FOR_ALL_ZEROS** (ie, waiting for the switch to become fully/stably active), **JUST_GONE_ACTIVE**, **WAITING_FOR_ALL_ONES** (waiting for the switch to become fully/stably inactive), and **JUST_GONE_INACTIVE**.

We can shorten these names to **WFAZ**, **JGA**, **WFAO** and **JGI**, respectively.

Remember that a copy of the state machine shown in Fig.2(a) is applied to each switch. The **setup()** function preloads the shift register (SR) associated with each switch with 0xFFFF and sets its state to **WFAZ**, reflecting the assumption that all our switches commence in their inactive states.

If you look at our **loop()** function, you'll see that once every millisecond, it first calls the **UpdateSwitches()** function, followed by the **ProcessSwitches()** function.

The **UpdateSwitches()** function handles noise rejection and switch debouncing. It's also responsible for transitioning the states from **WFAZ** to **JGA** and from **WFAO** to **JGI**. Meanwhile, the

ProcessSwitches() function performs any game-related actions (GRAs) associated with the switch being pressed or released.

Once these actions have been performed, this function is also responsible for transitioning the states from **JGA** to **WFAO** and from **JGI** to **WFAZ**.

When you come to a fork…

Yogi Berra (1925-2015) was an iconic American baseball player—catcher, manager, and beloved cultural figure. He was famous for his paradoxical and humorous "Yogi-isms", memorable sayings that appeared nonsensical at first, but often made profound sense the more you thought about them, such as "It's tough to make predictions, especially about the future".

One of my favourites is, "When you come to a fork in the road, take it". As with many of Yogi's Zen-like offerings, there's more to this than meets the eye. The most common reading is: don't overthink decisions; just choose a path and move forward. However, there is a backstory here.

When approaching Yogi's house, the road split into two branches, but both forks eventually led to the same destination, so it didn't actually matter which you chose. I tell you, you won't find nuggets of knowledge and tidbits of trivia like this in other, lesser electronics magazines!

The reason I mention this here is that, like a thief in a cutlery store, I feel the urge to take a fork. That's not something I expected to hear myself say when I woke up this morning.

To be slightly more specific, in software development, a 'fork' is an independent copy of an existing codebase created to explore or evaluate changes without affecting the main line of development. That's what we're going to do here.

Can you repeat that?

I don't want to incorporate auto-repeat into our main codebase at this time, so I've decided to take a fork. Our starting point will be our original flashing pixel program (**CB-apr26-code-02.txt**), which uses the RYGB action buttons to modify the colour of a flashing pixel and the NSEW direction buttons to move it around our 14 × 10 array. I've just forked this code into a test program named **CB-apr26-code-06.txt**.

In our original sketch, as depicted in Fig.2(a), each switch uses a simple four-state debounce state machine that generates one action-triggering event per press. This event occurs when the input becomes stably active and the machine enters its **JUST_GONE_ACTIVE** state. In the case of the action buttons, the event changes the flashing pixel's colour, while the direction buttons modify its location.

To add auto-repeat functionality to the direction buttons, we augment the per-switch state machines, as shown in Fig.2(b). First, we extended the state machine by adding a repeat counter, **rptCtr** (labelled RC in the diagram), to the **SwitchStuff** structure.

We also need to introduce a new **WAITING_FOR_NOT_ALL_ZEROS** (aka **WFNAZ**) state, which we can think of as 'waiting for the switch to stop being active and commence its transition to becoming stably inactive'. We'll use this state to distinguish between the button being held down and it being released.

The only change to the **UpdateSwitches()** function is that, once a switch becomes stably active, the code sets **rptCtr** to its initial delay value. We then enter the **JUST_GONE_ACTIVE** state, as before.

When the **ProcessSwitches()** function sees the **JUST_GONE_ACTIVE** state, it first performs any game-related actions (GRAs) associated with that switch. For the action switches, which don't require auto-repeat (at least, not in this instance), the state machine continues to transition to the **WAITING_FOR_ALL_ONES** state, as shown in Fig.2(a).

By contrast, for the direction switches, it transitions to our new **WAITING_FOR_NOT_ALL_ZEROS** state. While the state machine remains in that state and the button is still pressed, the code decrements **rptCtr** on each tick. When the counter expires (reaches zero), the code reloads **rptCtr** and generates another **JUST_GONE_ACTIVE** event, resulting in repeated movement in the selected direction.

As soon as the button is released, the state machine transitions to **WAITING_FOR_ALL_ONES**, after which it proceeds exactly as before.

I just ran a quick test, and this works rather well. Pressing any of the action buttons changes the flashing pixel's colour as before (pressing and holding the action buttons has no effect because we haven't modified their state machines). Pressing and releasing any of the direction buttons moves the flashing pixel one step in that direction, while pressing and holding them keeps it moving until the button is released.

As I noted earlier, I'm not sure if we will ever need this functionality, but now we know we can do it if we want to.

Timeless classics

Snake is an arcade and early computer game that emerged in the 1970s and has been continuously reinvented ever since. It belongs to a small, illustrious club of classic games, alongside titles like Pong and Tetris, that prove you don't need fancy graphics or complex rules to create something engaging.

Snake's inclusion on Nokia mobile phones in the late 1990s made it one of the most widely played games in history. The rules are instantly understandable (move, eat, grow, don't crash), yet the gameplay remains compelling. Snake still appears today in browsers, embedded systems demos, coding tutorials, and even as Easter eggs.

At its heart, Snake is beautifully simple: a series of connected segments move across a grid, one step at a time, changing direction under the user's control, growing longer as the game progresses and demanding ever more careful control.

This simplicity makes it perfect for newbie programmers because every part of the game—position, direction, movement, and state—cleanly maps onto fundamental programming concepts.

In our case, we'll bring this timeless concept to life on the 14 × 10 array of tricolour LED 'pixels' on our retro game console. We'll take this step-by-step (or slither-by-slither), starting by creating a short, four-segment snake at a known location. Later, we'll learn how to make it move, turn, and eventually (in later stages) hunt for food and grow. But first, we need to set the scene…

Circular thinking

Let's suppose that, at some time in the future, we end up with an 11-segment snake, as illustrated in Fig.3. One segment will act as the head (we'll colour this purple), with the remaining 10 segments acting as the body (we'll colour them green).

From Fig.3, we can deduce that, over the past couple of moves, the snake has been heading east. There are only three possible options for its next move: to carry on moving east, or to change direction and head north or south. It can't move west because that would mean moving back into itself.

Although the representation in Fig.3 appears simple, it hides underlying

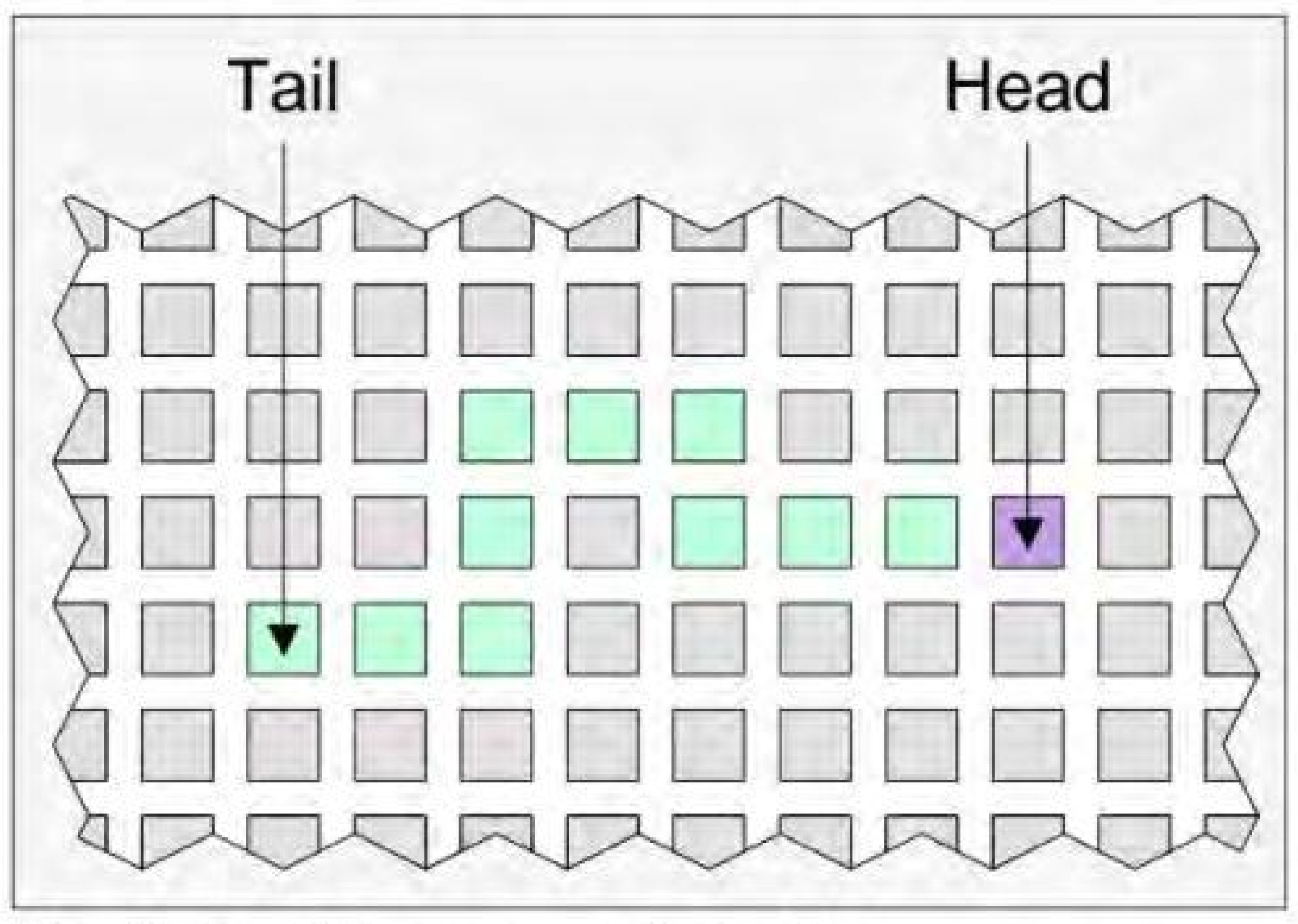


Fig.3: an 11-segment snake.

```
138  #define TICK                 1
139  #define SNAKE_MOVE_TICKS   500
140
141  #define NUM_SEGMENTS       140
142  #define START_HEAD           3
143  #define START_TAIL           0
144  #define START_LENGTH         4
145  #define START_LEVEL          1
146
147  #define NUM_DIRECTIONS       4
148  #define NORTH                0
149  #define SOUTH                1
150  #define EAST                 2
151  #define WEST                 3
152
153  #define HEAD_COLOR     MAGENTA
154  #define BODY_COLOR       GREEN
155
156  typedef struct
157  {
158    uint8_t x;
159    uint8_t y;
160  } SnakeSegment;
161
162  SnakeSegment Snake[NUM_SEGMENTS];
163
164  int16_t SnakeHead;
165  int16_t SnakeTail;
166  int16_t SnakeLength;
167  uint8_t SnakeDirection;
168  uint8_t GameLevel = START_LEVEL;
169
170  uint32_t TimeThen = micros();
171  uint32_t TimeNow;
172  uint32_t SnakeThen = TimeThen;
```

Listing 7(a): snake-related stuff.

layers of complexity. For example, although we think of our 'play area' as a 2D array, the pixels (LEDs) themselves are wired as a single long serial string. This means that snake segments that occupy adjacent squares on the grid might be dozens of pixels apart in the underlying 1D string.

Things become more complicated when we want the snake to advance. Fortunately, we don't have to redraw every segment. In the absence of any "food" (the "eating" of which will make the snake grow longer), moving the snake simply involves placing the head segment in its new position, changing the old head segment into a body segment, and deleting the tail segment.

On the downside, using a 2D array that ultimately maps onto a 1D string to track the segment locations can be a pain. The solution is to use a circular buffer. As a worst-case scenario, our snake can grow so large that it fills the grid, so we will create a circular buffer of 14 × 10 = 140 elements, numbered from 0 to 139, each capable of storing the X-Y location of one of the snake's segments.

Let's imagine the 11-segment snake from Fig.3 being stored somewhere in a 140-element circular buffer, as depicted in Fig.4. Once we've generated the initial snake, stored it (in the circular buffer), and drawn it (on the array), all we need to do is keep track of the locations of the head and tail in the circular buffer.

These can be represented as integers (I think of them as pointers), which we use to index into the buffer. Now, when the snake advances, in addition to updating the colours of any related pixels on the array, all we need to do is increment the head and tail pointers into the buffer. The reason we call this a circular buffer is that we manipulate it so that its end 'wraps around' to its beginning.

We will also keep track of the snake's length, which will eventually form the score in our game. The longer the snake, the higher the score.

When you think about it, if we know the value of the head pointer and the length of the snake, then we can easily generate the value of the tail pointer on-the-fly whenever we feel the need. Personally, however, I prefer to maintain both head and tail pointers (it's my 'programmer's prerogative').

Hello, snake!

Traditionally, a programmer's first tentative step when learning a new language or working with a new system is to coax the computer to print some variant of "Hello, world!" (the original *C Programming Language* book used "hello, world" without capitalisation or an exclamation mark).

The underlying idea is both modest and powerful: to prove that the toolchain works, confirm that text can be produced, and celebrate the smallest possible success before attempting anything more ambitious.

We are going to do something similar with our Snake program. We'll start by generating a four-element snake, storing it in our circular buffer, and drawing it on our array.

I just copied our fundamental framework (named **CB-apr26-code-01.txt**) and made a few tweaks to implement a "Hello, snake!" program (in the file named **CB-apr26-code-07.txt**).

Our first task is to add game-related definitions and declarations, as illustrated in Listing 7(a). We still have our 1ms **TICK** definition on line 138, to which we add a 500ms (half-second) **SNAKE_MOVE_TICKS** definition on line 139 (we won't be using this in this program, but it's there for when we need it).

On lines 141 through 145, we define the number of elements in our circular buffer (140), the values of the snake's head and tail pointers into the buffer (3 and 0), the length of the snake (4 segments) and the game's starting level (1).

On lines 147 through 151, we define the number of directions the snake can move (4) and associate values of 0, 1, 2, and 3 with directions north, south, east and west, respectively.

On lines 153 and 154, we define the colours associated with the snake's head and body segments.

On lines 156 through 160, we define a structure to store the X-Y coordinates associated with an individual snake segment. Then, on line 162, we instantiate our circular buffer as a 140-element array of these structures.

Finally, on lines 164 through 172, we declare the variables that hold the snake's current state—head and tail indices (pointers) into the circular buffer, the snake's length and direction, and the current game level—along with a couple of timing variables we'll use to schedule events like snake motion without resorting to blocking delays.

If you look at the **setup()** function in **CB-apr26-code-07.txt**, you'll see that we've added calls to two new functions at the end, just after the calls to our power-on self-test functions. The first, **InitializeSnake()**, generates our 4-element snake, stores it in our circular buffer, and draws it on our array. For simplicity, we're starting with a horizontal snake pointing east (Fig.5).

The second, **InitializeDisplays()**, sets up our 7-segment displays, presenting "L" (for level) in the most significant digit (digit 5), the current game level (1) in digit 4, and the current length of the snake (4) in the three least significant digits (2, 1, and 0).

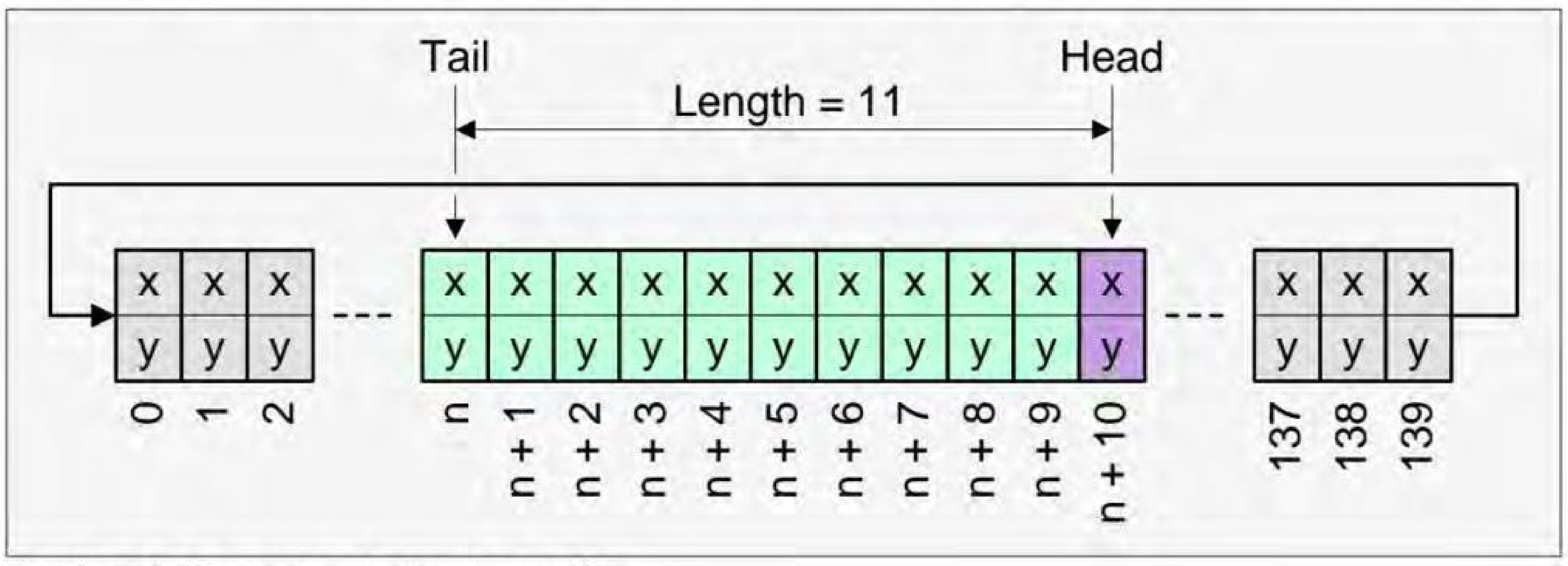


Fig.4: a 140-element circular buffer.

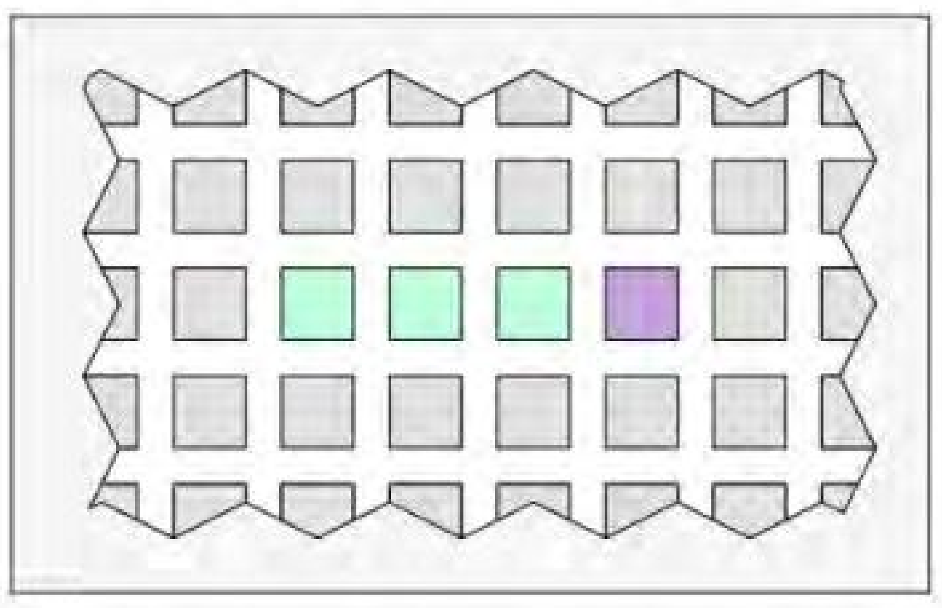


Fig.5: hello there, snake!

I just ran this program, and it looks very… static, but at least we've laid the groundwork for what's to come.

A simple slither

Now, let's make our snake a little more animated. We'll start with a simple 'slither' by just making it move eastward. Furthermore, when the snake reaches the right-hand side of the array, it will 'wrap around' and reappear on the left-hand side.

First, in the file **CB-apr26-code-08.txt**, we modify our loop() function, as seen in Listing 8(a). The only change here appears on line 217, where we call a new UpdateSnake() routine. It's worth noting that we are still calling the functions to debounce and process our switches (lines 218 and 219, respectively); it's just that we aren't actually doing anything with them yet, but we will…

Next, we add the new UpdateSnake() routine itself, as illustrated in Listing 8(b). One thing you'll notice is that the controlling construct on line 474 is very similar to the one we used in the UpdateFlash() function in our Flashing Pixel and Etch A Sketch programs.

The way this is currently set up, with SNAKE_MOVE_TICKS defined as 500ms, is that the snake will move every half-second. One thing we might do in the future is to gradually decrease this value, thereby speeding up the snake and increasing the difficulty in controlling it as we progress through the game's levels.

On line 476, we determine the location of the new head pointer for use with our circular buffer. We've seen this use of the modulo (%) operator before. In simple terms, this modulo operation makes the buffer behave as if its ends are joined together.

```
210  // Do this over and over again
211  void loop()
212  {
213    TimeNow = millis();
214
215    if ( (TimeNow - TimeThen) >= TICK)
216    {
217      UpdateSnake();
218      UpdateSwitches();
219      ProcessSwitches();
220
221      TimeThen = TimeNow;
222    }
223  }
```

Listing 8(a): tweaking the loop() function.

Each time we advance the head pointer, it increments by one. When it reaches the last element, the modulo calculation quietly wraps it around to zero so we can continue from the start without any special-case code.

In the next iteration of the code, we're going to allow the snake to move in any direction. For the moment, however, we're assuming it always wants to move east, which explains lines 478-486.

Observe the use of the modulo operators on lines 488 and 489. These allow the snake to wrap around the array in any direction, east to west and north to south. The reason we've added + NUM_COLS in line 488 and + NUM_ROWS in line 489 is to prevent problems if deltaX is -1 or deltaY is -1, respectively.

I've just run this incarnation of the program, and it's a wonder to behold, but it leaves me wanting more.

Put your snake on automatic

If I had a coconut for every time I've heard someone say, "Put your snake on automatic", I wouldn't have many coconuts, but that's not important here.

What we're going to do now is allow our snake to move randomly around the array. First, however, let's return to the rationale for the order in which we perform the actions of removing the tail segment before moving the head segment, as shown in Listing 8(b), lines 495-496 and 498-502, respectively.

There are two ways our snake could move. It could stretch out its head and then pull in its tail, or it could pull in its tail and then stretch out its head. We've chosen to implement the latter. This reflects the behaviour of the classic Snake game, where the head can move into the square the tail has just vacated.

```
466  void UpdateSnake ()
467  {
468    int16_t tmpSnakeHead;
469    uint8_t tmpX;
470    uint8_t tmpY;
471    uint8_t deltaX;
472    uint8_t deltaY;
473
474    if ( (TimeNow - SnakeThen) >= SNAKE_MOVE_TICKS)
475    {
476      tmpSnakeHead = (SnakeHead + 1) % NUM_SEGMENTS;
477
478      // Generate new X-Y location for the head
479      // This part coming soon
480
481      // Let's asume the direction is east
482      if (SnakeDirection == EAST)
483      {
484        deltaX = 1;
485        deltaY = 0;
486      }
487
488      tmpX = (Snake[SnakeHead].x + NUM_COLS + deltaX) % NUM_COLS;
489      tmpY = (Snake[SnakeHead].y + NUM_ROWS + deltaY) % NUM_ROWS;
490
491      // Remove tail segment
492      DrawPixel(Snake[SnakeTail].x, Snake[SnakeTail].y, BLACK);
493      SnakeTail = (SnakeTail + NUM_SEGMENTS + 1) % NUM_SEGMENTS;
494
495      // Set old head segment to body color
496      DrawPixel(Snake[SnakeHead].x, Snake[SnakeHead].y, BODY_COLOR);
497
498      // Move head and set it to head color
499      SnakeHead = tmpSnakeHead;
500      Snake[SnakeHead].x = tmpX;
501      Snake[SnakeHead].y = tmpY;
502      DrawPixel(Snake[SnakeHead].x, Snake[SnakeHead].y, HEAD_COLOR);
503
504      Neos.show();
505
506      SnakeThen = millis();
507    }
508  }
```

Listing 8(b): implementing a simple eastward slither.

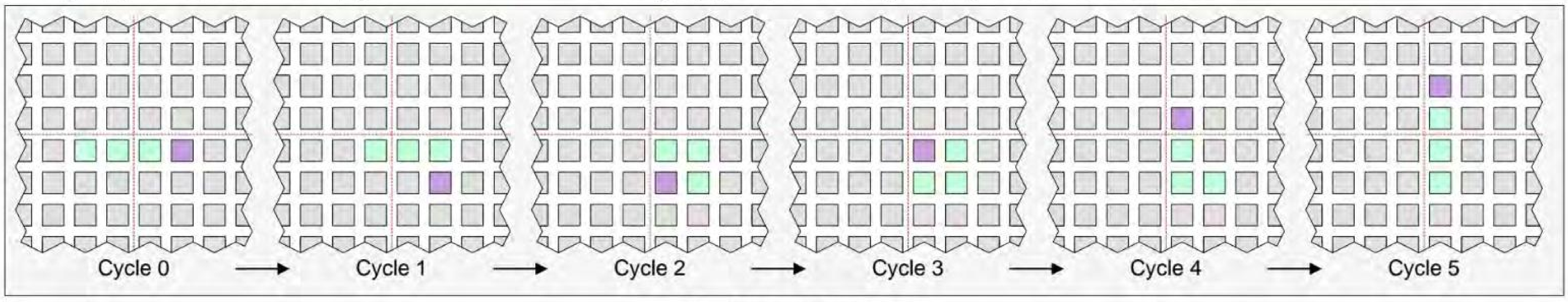


Fig.6: don't eat my tail!

```
479      // Pick a new direction (can't be 180 degrees)
480      do {
481        tmpDirection = random(0, NUM_DIRECTIONS);
482      } while (tmpDirection == (SnakeDirection ^ 1));
483
484      SnakeDirection = tmpDirection;
485
486      // Generate new X-Y location for the head
487      if (SnakeDirection == NORTH)
488        {deltaX =  0; deltaY =  1;}
489      else if (SnakeDirection == SOUTH)
490        {deltaX =  0; deltaY = -1;}
491      else if (SnakeDirection == EAST)
492        {deltaX =  1; deltaY =  0;}
493      else if (SnakeDirection == WEST)
494        {deltaX = -1; deltaY =  0;}
495
496      tmpX = (Snake[SnakeHead].x + NUM_COLS + deltaX) % NUM_COLS;
497      tmpY = (Snake[SnakeHead].y + NUM_ROWS + deltaY) % NUM_ROWS;
```

Listing 9(a): moving in a random direction.

By removing the tail first, we avoid falsely detecting a collision with the snake's own body and allow the familiar smooth, continuous motion players expect. It's only when the snake eats food (which we'll implement next month) that we skip the tail-removal step, thereby causing the snake to grow by one segment.

All this is depicted in Fig.6. Our snake starts by progressing eastward (Cycle 0). Then it turns south (Cycle 1), then west (Cycle 2), then north (Cycle 3). This is the key point in the snake's journey. If we didn't remove the tail segment before drawing the new head segment, our snake would effectively try to eat its own tail, which would be a bad thing to happen (ask a real snake if you don't believe me).

The only changes we need to make to implement the snake's random motion in the file named **CB-apr26-code-09.txt** are illustrated in Listing 9(a). The first thing we do on lines 480-482 is loop around, generating a new random direction (0, 1, 2 or 3), corresponding to north, south, east and west, respectively.

The rule we need to implement here is that the snake cannot 'go back on itself'. To put this another way, if it's heading north, it cannot decide to head south (and vice versa). If it's heading east, it cannot choose to head west (and vice versa).

The key observation is that opposite directions form pairs: north ↔ south (0 ↔ 1) and east ↔ west (2 ↔ 3). This means each pair differs by only the least significant bit.

Using the XOR operator (**^**) in the expression **SnakeDirection ^ 1** in the conditional test on line 482 performs a delightful bit of digital sleight-of-hand, flipping the least-significant bit of the current direction to swap north with south and east with west. If the random choice matches this forbidden opposite, we simply try again until a legal direction appears.

Once we've determined the snake's new direction, we use lines 487-494 to set up appropriate **deltaX** and **deltaY** values, which we then use to generate the new X-Y location of the snake's head.

I don't know about you, but I feel a little "Tra-la" is in order. As I pen these words, my four-segment snake is happily roaming around its little universe.

Next time

Our next task will be to take control of the snake's path using our direction buttons. We'll also start randomly adding food, the eating of which will make our snake grow longer.

We will keep adding games and performing experiments with our console in the months to come. At the same time, we will be starting a new project that's so exciting I can barely contain myself.

As always, if you have any thoughts you'd care to share, please feel free to drop me an email at max@clivemaxfield.com PE

PoLabs
For more information or software download please visit
www.poscope.com/epe

PoKeys
Connect, Control
- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers
Drive
- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+
PoScope Mega50
Measure
- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

Circuit Surgery

Regular clinic by Ian Bell

Measuring circuit frequency response using a PC sound card, part 7: op amp output impedance & compensation

In recent articles, we have been looking at frequency response measurement. This was motivated by the need to measure the response of a practical implementation of an example digital filter (from our series on DSP) using the sound card (audio interface) of a PC or laptop together with software for this purpose.

As part of this topic, we have been discussing the signal conditioning needed to interface the circuit under test to the sound card's line inputs and outputs. Most recently, we focused on op amp based amplifier feedback, frequency response and stability.

Last month, we covered the effect of capacitive loading on op amp amplifiers and its potential to cause instability (unwanted oscillation). This is relevant to the signal conditioning circuit, but our discussion is applicable to a wide range of other applications.

This month, we will continue on this theme, looking at some aspects we did not cover last month, including how we might mitigate the effect of capacitive loads on stability.

Op amp stability recap

Op amp based amplifiers rely on negative feedback. The op amp on its own (without feedback) has an open-loop gain, A_O. The amplifier circuit with feedback has a closed-loop gain, A_C. Its loop gain is $-\beta A_O$; this is the gain around the closed feedback loop, where β is the fraction of the output that is subtracted from the input signal.

The frequency response of op amps (as shown in Fig.1) is important when discussing circuit stability. As signal frequency increases, the phase shift due to the delay around the loop becomes more negative, so the effective total phase shift reduces from 180° towards (and then past) zero.

If the total phase shift is zero or an integer multiple of ±360°, the feedback will be positive, potentially resulting in instability. Phase margin (see Fig.1) and gain margin are measures of how close a circuit is to being unstable.

The open-loop gain of most op amps is deliberately reduced at high frequencies so that unity-gain amplifiers are stable. This typically creates a low-frequency (typically 1-10Hz) pole in the open-loop response (Pole 1 in Fig.1 is typical of an op amp dominant pole). Above this frequency, the open-loop gain falls from A_O at 20dB per decade until another pole or zero is reached.

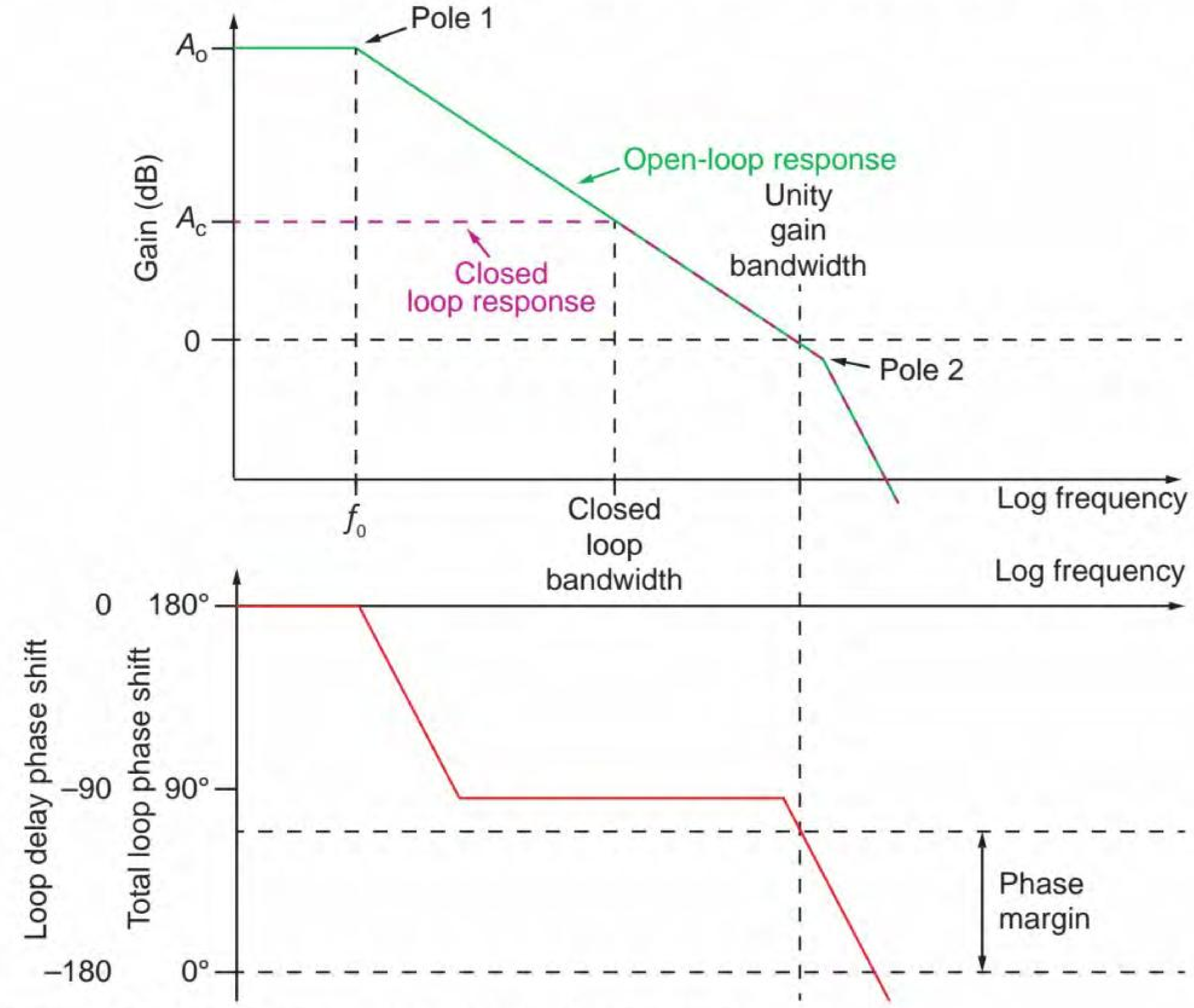


Fig.1: the typical frequency response of an op amp.

Capacitive loading

Last month, we performed a simplified analysis of a unity-gain buffer with a capacitive load (Fig.2). We will reuse this model later, so it is worth recapping.

The circuit includes the op amp's output impedance (Z_O) because its interaction with the load capacitance is an important factor in circuit stability. To keep the analysis simple, we assume the op amp output impedance is resistive ($Z_O = R_O$) and its input impedance is infinite. We also ignored the resistive part of the load (R_L in Fig.2).

We also assumed an idealised op amp with just one low-frequency pole in its open-loop frequency response, $A_O(s)$, Pole 1 in Fig.1, and no frequency dependence of the direct feedback connection (β = 1 at all frequencies). The result was that the capacitive load added a pole to the loop-gain at a frequency of $f_p = 1 \div 2\pi R_O C_L$.

In this scenario, the circuit must be stable without the capacitive load because the single pole in $A_O(s)$ only provides -90° of phase shift. With the additional

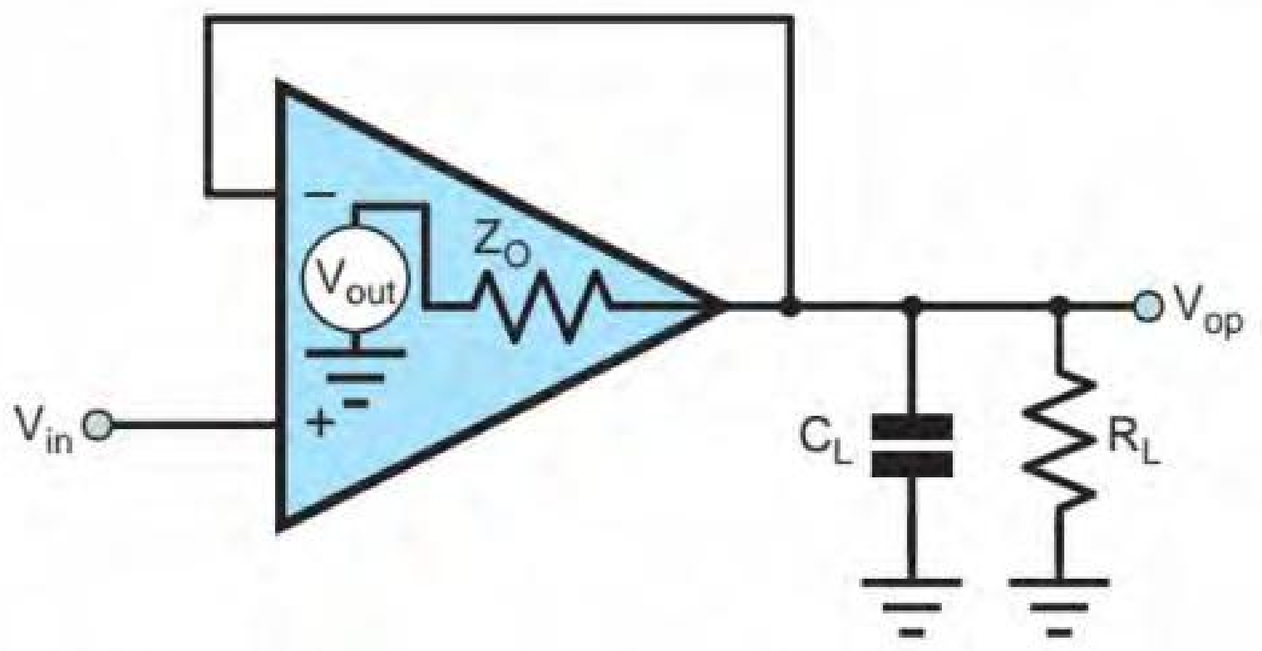


Fig.2: an op amp based buffer with a capacitive load.

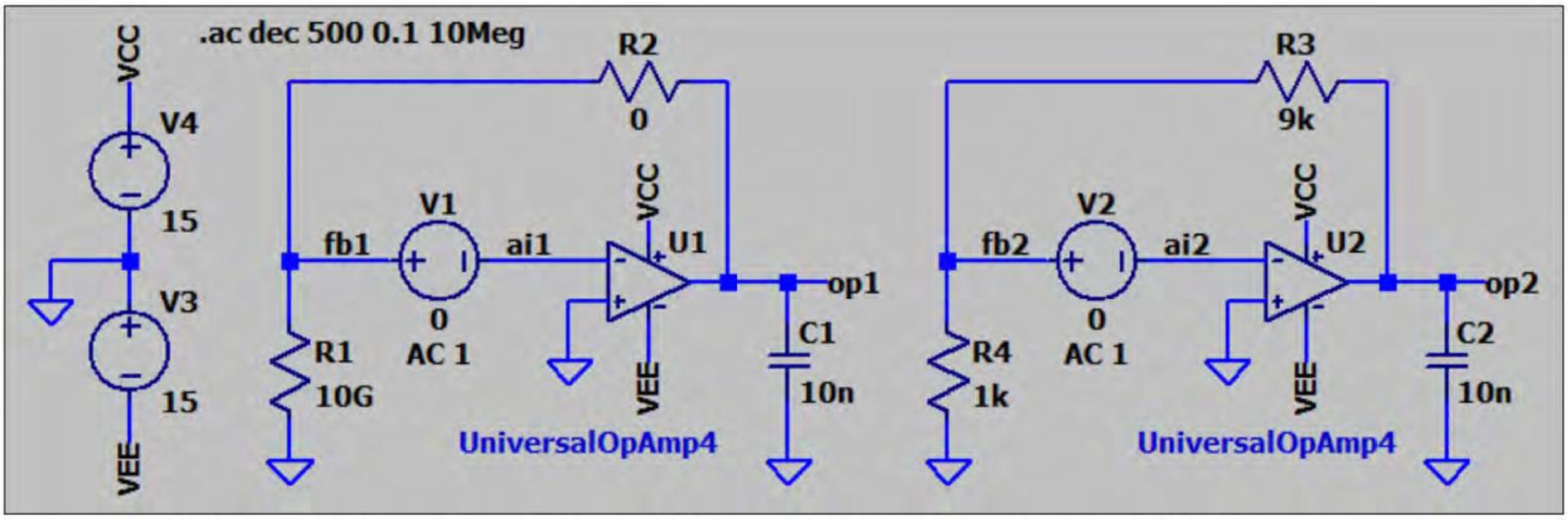


Fig.3: an LTspice schematic for simulating the loop gain of unity and higher-gain amplifiers based on op amps.

pole provided by R_O and C_L, the phase shift can reach 180° and potentially cause instability.

Real op amps have more than one pole (as shown in Fig.1), so the additional pole due to the capacitive load can easily cause sufficient phase shift for instability under certain conditions. Simulations using the OPAx134 op amp from the sound card interface showed marginal stability with a 1nF load and full instability with 2nF.

Amplifier gain

There are a couple of things that we did not include in our discussion last month that are worth looking at now. We only considered unity-gain circuits, and we assumed the output impedance of the op amp was resistive. We will continue to use this latter simplification in further examples, but it is worth looking at what a more realistic view of output impedance looks like.

First, though, we will consider the relationship between closed-loop gain and stability.

Fig.3 shows an LTspice schematic for simulating loop-gain for two op amp circuits, one with 100% feedback (using U1) and the other (using U2) having a feedback factor of $\beta = 0.1$, set by the potential divider ratio of R3 and R4, that is R4 ÷ (R3 + R4) = 1kΩ ÷ (1kΩ + 9kΩ) = 0.1.

This simulation is of loop gain, which determines stability, not the closed-loop gain of a specific amplifier circuit. Therefore, the circuits in Fig.3 do not have an input signal because we are just looking at the feedback loop.

U1 is configured as a unity-gain amplifier with its output connected directly to the inverting input (R2 has zero resistance). This is the circuit setup we used last month.

For U2, the feedback configuration could relate to a non-inverting amplifier with a gain of 10, or an inverting amplifier with a gain of 9. The gain of the non-inverting amplifier is found from $1 \div \beta = 1 \div 0.1 = 10$, or we can use the well-known resistor formula: 1 + R3 ÷ R4 = 1 + 9kΩ ÷ 1kΩ = 10.

The gain of the inverting amplifier is found from $1 - (1 \div \beta) = 1 - (1 \div 0.1) = -9$, or we can use the other well-known resistor formula: -R3 ÷ R4 = -9kΩ ÷ 1kΩ = -9. For more detail on the relationship between circuit gain and β, please refer to the February 2026 *Circuit Surgery* column.

The simulation setup in Fig.3 is similar to the simulation of idealised unity-gain amplifiers last month. It uses the UniversalOpAmp4 model with $A_O = 1.778\times10^6$ (125dB) and GBP = 8MHz, but with the output resistance R_O set to a more idealised value of 10Ω to reduce interaction with the resistor network.

Fig.4 shows the results from the simulation in Fig.3. The gain curves show that the loop gain for the higher gain (×10 or ×-9) amplifier (yellow trace) is 20dB below the loop gain of the unity-gain amplifier. Note the reciprocal relationship between closed-loop gain and loop gain – lower loop gain corresponds with higher closed-loop gain.

At DC, and frequencies well below the op amp's dominant pole, the loop gain of the unity-gain amplifier is βA_O, with $\beta = 1$, the loop gain is equal to A_O, which was set at 125dB in the op amp model, as noted above. For the higher-gain amplifier, we have $\beta = 0.1$, so the loop gain is 1/10th of the loop gain of the unity-gain circuit. In decibels, 1/10th is a difference of $\log_{10}(1/10) = -20$, so we expect a DC loop gain of 125dB – 20dB = 105dB, which is what we see in Fig.4.

The graph in Fig.4 has been annotated with the phase margin and shows that the phase margin of the higher gain circuit (65°) is much better than that of the unity-gain buffer (23°). Fig.4 shows that the phase response is the same for both circuits. This is because the two feedback networks are purely resistive, so neither adds any delay (and hence phase shift) to the loop.

However, as just discussed, the higher closed-loop gain circuit has a feedback potential divider that reduces the loop gain. This means the loop gain of the higher-gain amplifier falls to unity (0dB) at a lower frequency (727kHz) than the unity-gain circuit (3.39MHz). The 0dB gain values are shown in the phase margin annotations.

Phase margin is the difference between 0° and the total loop phase shift (including 180° due to inversion) at unity loop gain (see Fig.1). Or we can express it as the difference in the phase shift due to

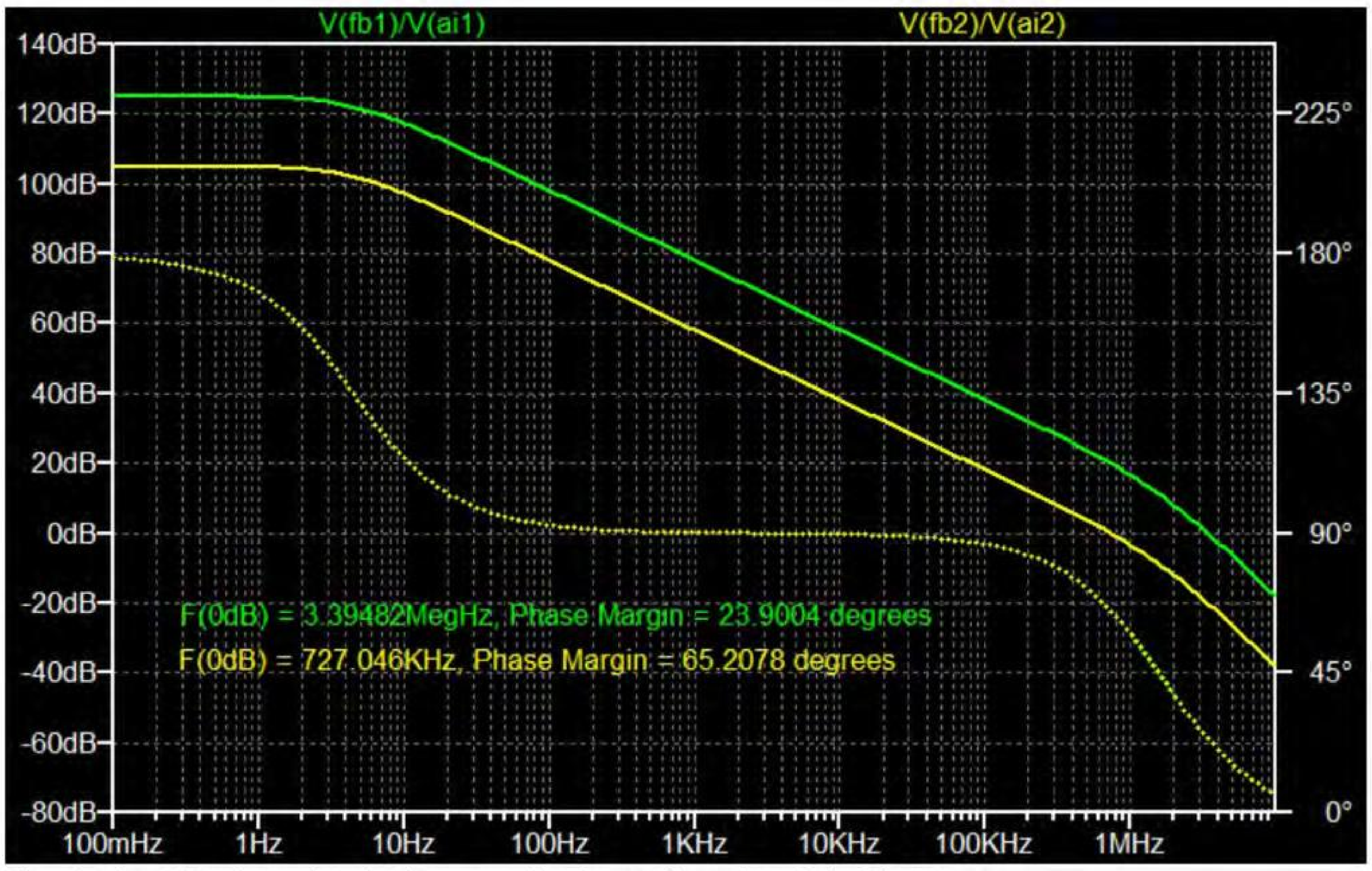


Fig.4: the LTspice simulation results from the circuit in Fig.3.

loop delay from -180° (also see Fig.1). The equal phase shift for the two circuits, and the lower loop-gain of the higher-gain amplifier, means that the phase margin of the higher-gain amplifier is better.

The phase margin measurements are illustrated in the zoom-in on the loop response plot shown in Fig.5.

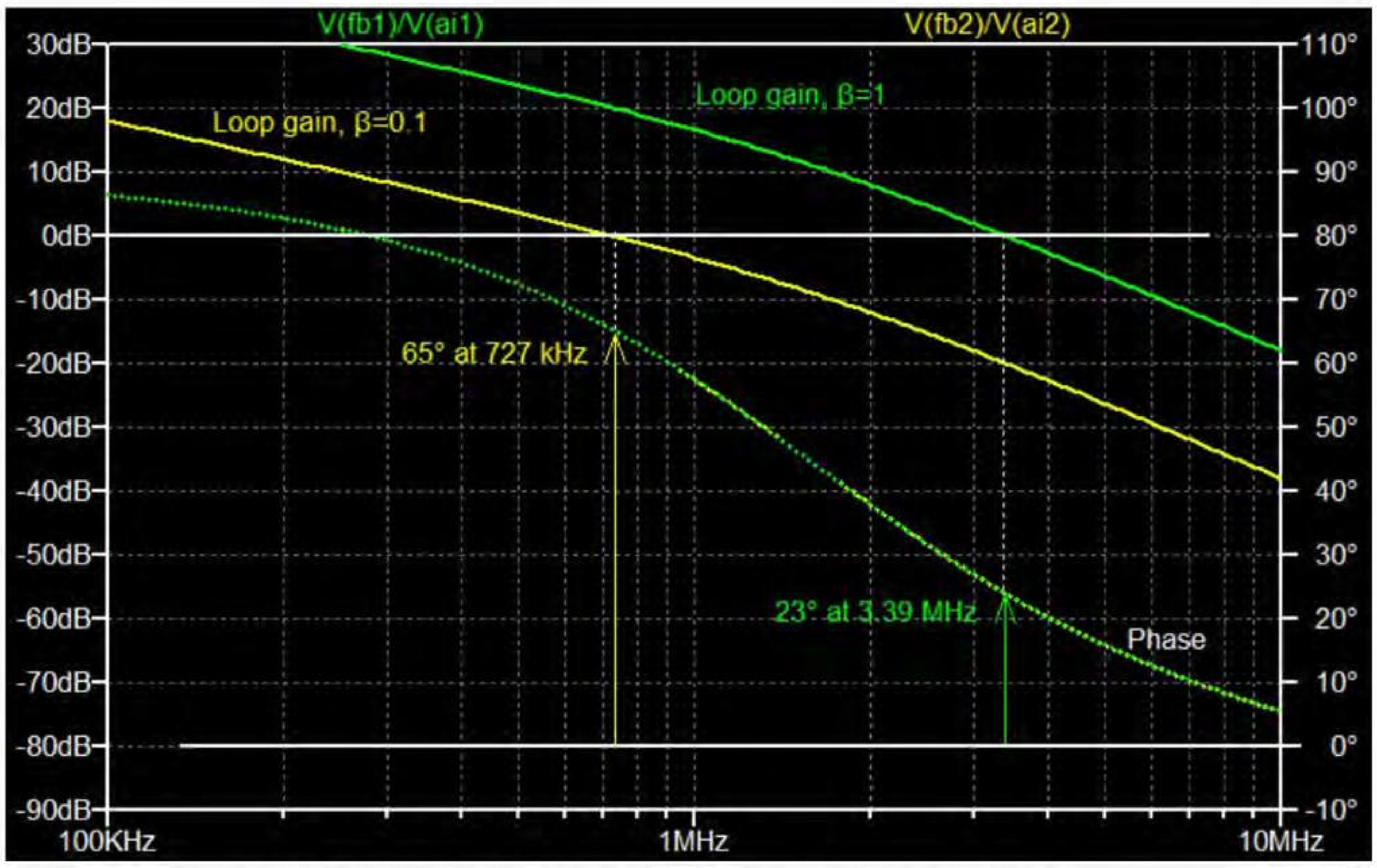


Fig.5: a zoom-in of Fig.4 showing the phase margin measurements.

As noted in our introduction to op amp frequency response in February 2026, the fact that lower closed-loop gain op amp amplifiers are more likely to be unstable may seem counterintuitive to some. Perhaps this is because people think of a high-gain circuit as being 'unstable' because it is very sensitive to changes in its inputs ignals, so may react in unwelcome ways if the input is too large.

It could also be because people often experience feedback from loudspeakers via microphones and musical instruments (known as acoustic feedback or the Larsen effect). In such a case, you are inside the loop, so turning down the amplifier (reducing gain) to stop the screeching sound is effectively reducing loop-gain, thus making the system more stable in line with the discussion above.

Reducing the gain of the amplifier here does not correspond to a lower-gain op amp circuit, but it is possible to see that it might be perceived as such.

It is also worth noting that the gain and phase shift responses of an acoustic scenario will typically be much more complex than an op amp based amplifier, due to multiple sound reflections and other room acoustic properties, which will change as people and microphones move.

Uncompensated op amps

The low-frequency pole (Pole 1 in Fig.1) found in most op amp open loop frequency responses is designed to ensure stability in unity-gain circuits built with the op amp. Modification of the frequency response to ensure stability is called compensation. These devices are referred to as compensated or 'unity-gain stable' op amps.

Some op amps are not designed to be stable in low closed-loop gain configurations and must be used in higher-gain configurations to be stable. These are referred to as decompensated op amps.

Other op amps are uncompensated – they do not have any compensation built in, so to be stable in any circuit, they must be compensated by the user adding external components; for example, a capacitor connected between two pins provided for that purpose. Decompensated op amps may also have pins to allow the user to increase the level of compensation.

Fig.6 is a comparison of the gain responses of unity-gain stable, decompensated and uncompensated op amps, assuming they all have the same DC open-loop gain (A_O).

The decompensated op amp has a low-frequency pole at a slightly higher frequency than the unity-gain-stable device, but has the same basic shape to the frequency response. The uncompensated op amp has it first pole at a higher frequency and a more complex-looking response because its high-frequency poles are more dominant in the shape of the graph.

Fig.6 is a little simplified as related devices with different compensation may not have exactly the same open-loop gain, and the high-frequency poles of the decompensated and uncompensated op amps are not shown.

Decompensated op amps (and uncompensated ones with suitable compensation added) allow closed-loop circuits to have higher bandwidth as long as they are stable at the required closed-loop gain (A_C), with all other things being equal.

This can be seen in Fig.6, where the closed-loop bandwidth of the decompensated device (dotted cyan trace) is larger than for the unity-gain stable device (dotted magenta trace).

Example devices

The NE5532 is a widely used low-noise dual op amp designed for audio use that is unity-gain stable (it would be suitable in place of the OPAx134 in the sound card interface). The NE5534 is a similar single (not dual) device with the same op amp circuitry as the NE5532, except that it is decompensated so that it is only stable for a closed-loop gain of three or more.

This allows the NE5534 to achieve higher bandwidth than the NE5532. The NE5534 also has a higher slew rate (maximum rate of change of the output), which allows it to handle faster transients and larger signals without distortion.

The frequency response of the NE5534 can be optimised to specific applications using an external compensation capacitor between the COMP and COMP/BAL pins. However, the datasheet only provides example compensation capacitor values on some characteristic tables and curves (some datasheets provide formulae for compensation capacitor values).

With a 22pF external compensation capacitor, the NE5534 is unity-gain stable and has the same unity-gain bandwidth as the NE5532. This makes sense because of the similarity of the devices.

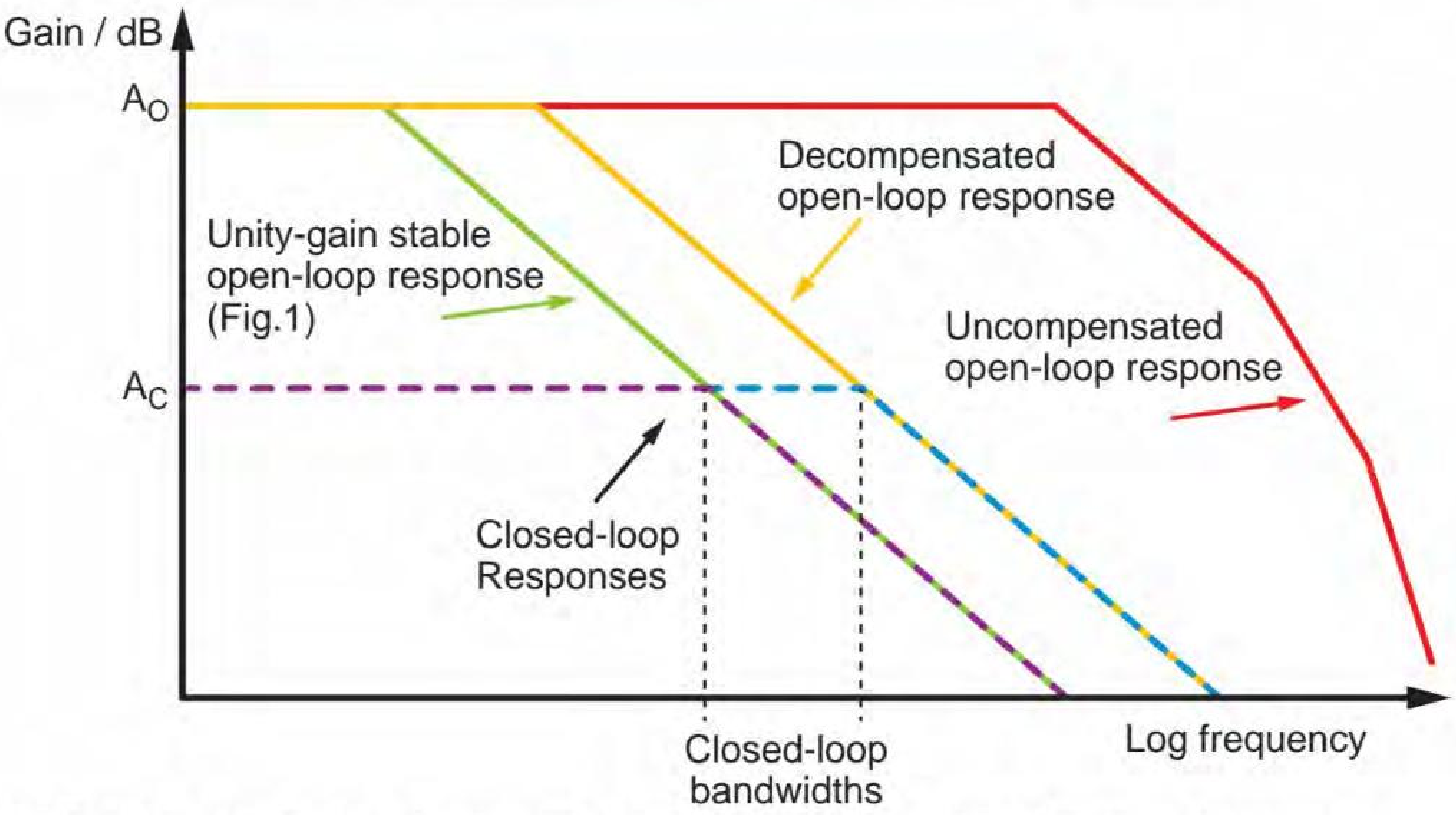

Gain / dB
A_O
A_C
Unity-gain stable open-loop response (Fig.1)
Decompensated open-loop response
Uncompensated open-loop response
Closed-loop Responses
Closed-loop bandwidths
Log frequency

Fig.6: unity-gain stable, decompensated and uncompensated op amp responses

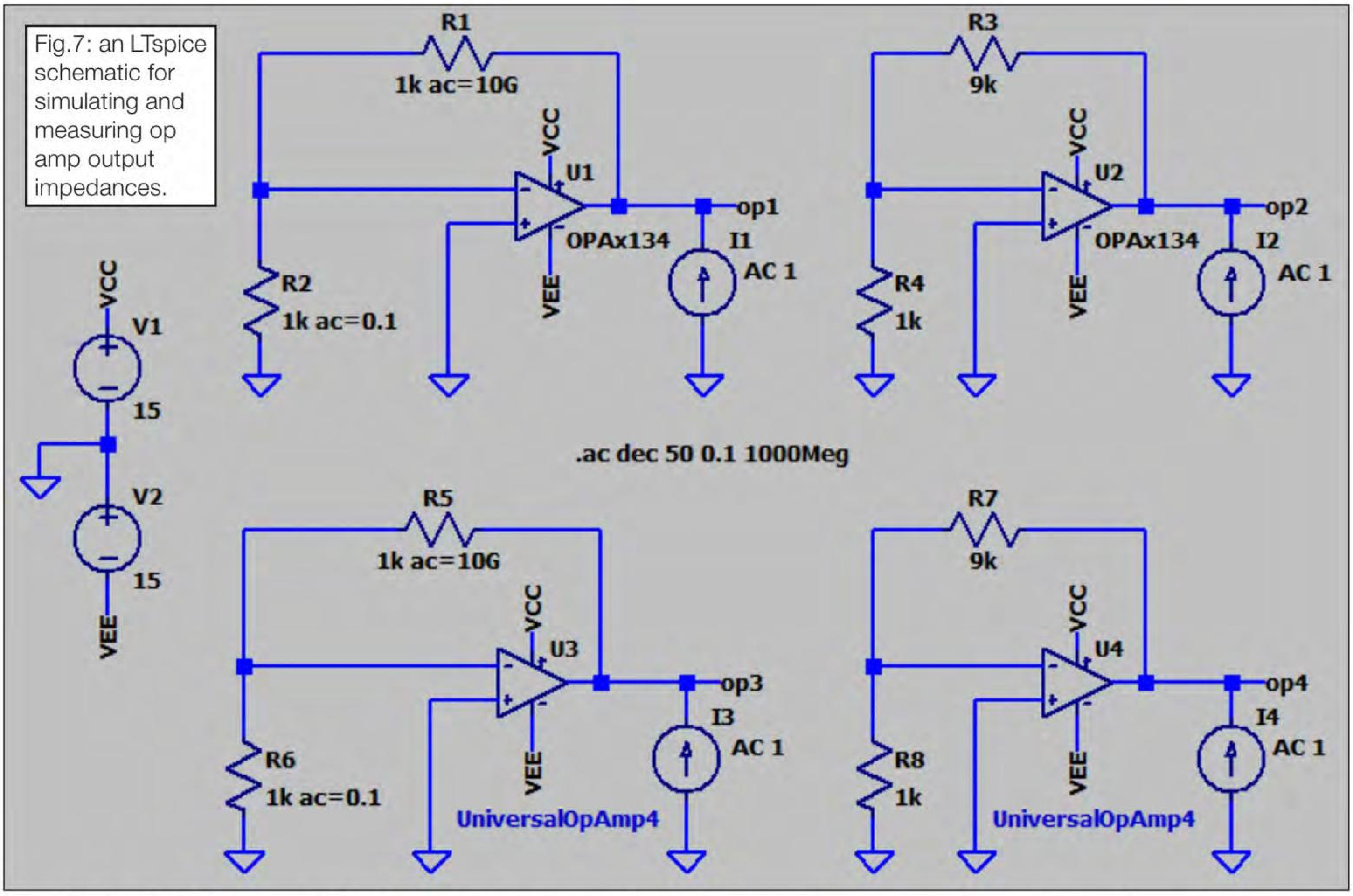


Fig.7: an LTspice schematic for simulating and measuring op amp output impedances.

Higher bandwidths are available from decompensated op amps in higher-gain circuits if they have not been further compensated to obtain unity-gain stability. In general, the extra complexity and need for external compensation capacitors mean they are less widely used than unity-gain stable devices.

Similarly, the OPAx227 low-noise op amps from Texas Instruments are unity-gain stable, while the OPAx228 are optimised for closed-loop gains of 5 or more. These devices are available in single, dual and quad packages and, unlike the NE5534, do not have compensation pins.

Op amp output impedance

We have assumed a fixed-resistance output impedance to simplify the analysis of op amp frequency response. However, the actual impedance is more complex and varies with frequency. We can obtain plots of op amp output impedance against frequency using LTspice.

One way to achieve this is to inject a current (*i*) into the op amp's output in an AC analysis and plot $v \div i$, where v is the voltage at the op amp's output. The op amp has zero input signal in this simulation.

An LTspice schematic for simulating op amp output impedance is shown in Fig.7. Both open-loop (U1,U3) and closed-loop (U2,U4) impedances are investigated. We look at both a real op amp (the OPAx134 – U1,U2) and an idealised case using the UniversalOpAmp4 model (U3,U4).

The UniversalOpAmp4 is configured as for Fig.3 except its output resistance is set to 200Ω.

The open-loop circuits use the approach to simulation we discussed in February 2026. We use separate DC and AC resistance values in the feedback network to ensure the op amp is biased correctly by applying feedback in the simulator's operating point calculation, but it is open loop for the AC analysis.

The results are shown in Fig.8, with the OPAx134 at the top and the idealised model below. The open-loop output impedance OPAx134 (green trace) varies significantly with frequency, whereas the idealised open-loop output impedance (yellow trace) is flat throughout the frequency range at the value set in the model parameter. 200Ω matches the low-frequency value from the OPAx134.

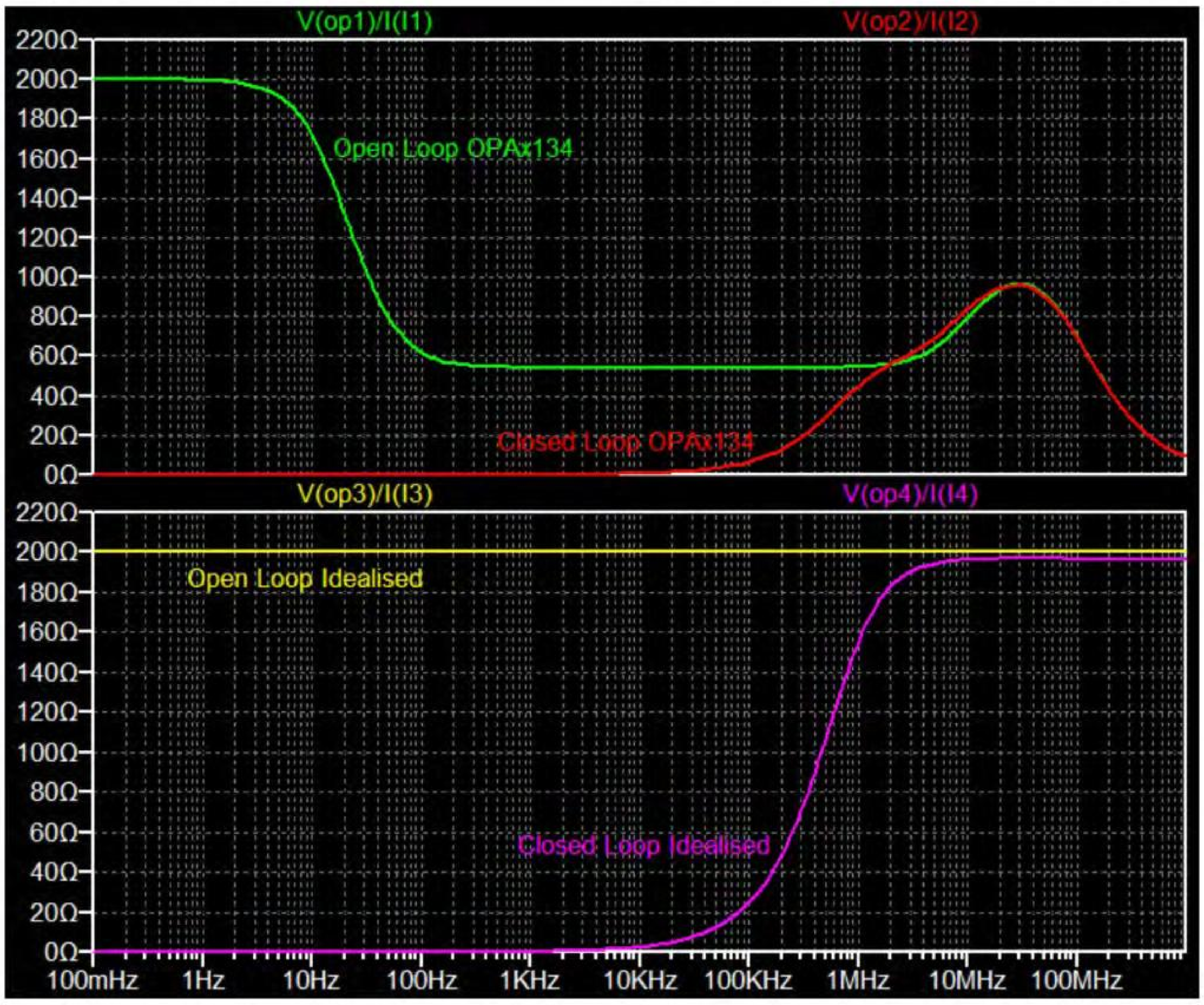


Fig.8: the results from running the simulation using Fig.7.

The shapes of the open-loop output impedance curves for different op amps vary, but the kind of variation shown for the OPAx134 in Fig.8 is not unusual.

Flat parts of the curve (low and mid-range frequencies) show resistive behaviour. The parts of the curve that slope down as the frequency increases exhibit capacitive behaviour (capacitor impedance drops with increasing frequency), whereas upward slopes are inductive.

The closed-loop output impedance in both cases (red trace for the OPAx134, magenta for the idealised op amp) is close to zero until higher frequencies are reached. In these circuits, the output impedance is reduced by $(1 + \beta A_O)$ with feedback applied.

For example, with $A_O = 125\text{dB} = 10^{6.25}$ and $\beta = 0.1$, we get an expected reduction by a factor of $(1 + \beta A_O) \approx 0.1 \times 10^{6.25} = 10^{5.25}$ (we can ignore the "1 +"). This gives an impedance of $200\Omega \div 10^{5.25}$ which is around 1mΩ. Zooming in on the plots confirms an impedance around this value. β is set by the feedback resistors for U2 and U4, as discussed for the circuit in Fig.3.

For both the OPAx134 and idealised cases, the impedance starts to noticeably increase in the plots at around 10-100kHz. In both cases, the curves then tend towards following the open-loop output impedance as the frequency increases.

The open-loop gain response of the OPAx134 is shown in Fig.9 (we discussed this in the February 2026 issue). This shows that by 100kHz, A_O has dropped to about 45dB, that is, 125dB – 45dB = 80dB lower than at DC. This significantly decreases $(1 + \beta A_O)$, so the factor by which the output impedance is reduced decreases accordingly.

80dB is a factor of 10^4, so we expect the output impedance to be about 10^4 larger than at DC; $1\text{m}\Omega \times 10^4 = 10\Omega$, which is in line with the impedance value on the graphs in Fig.8.

The fact that the beneficial effect of feedback reducing effective output impedance diminishes at high frequencies can impact circuit performance.

One relatively well-known example is in Sallen-Key low-pass filters, where the gain can rise at high frequencies, after falling above the cutoff frequency, due to increasing output impedance being unable to compensate for signal feedthrough in the capacitor between the input network and the op amp's output.

[Editor's note: I prefer multi-feedback topology filters as they do the same job with just one extra resistor and don't suffer from this problem to the same extent, although they are not immune.]

Whether the high-frequency impedance changes are important or not will depend on the specific details of each circuit and the signals being processed.

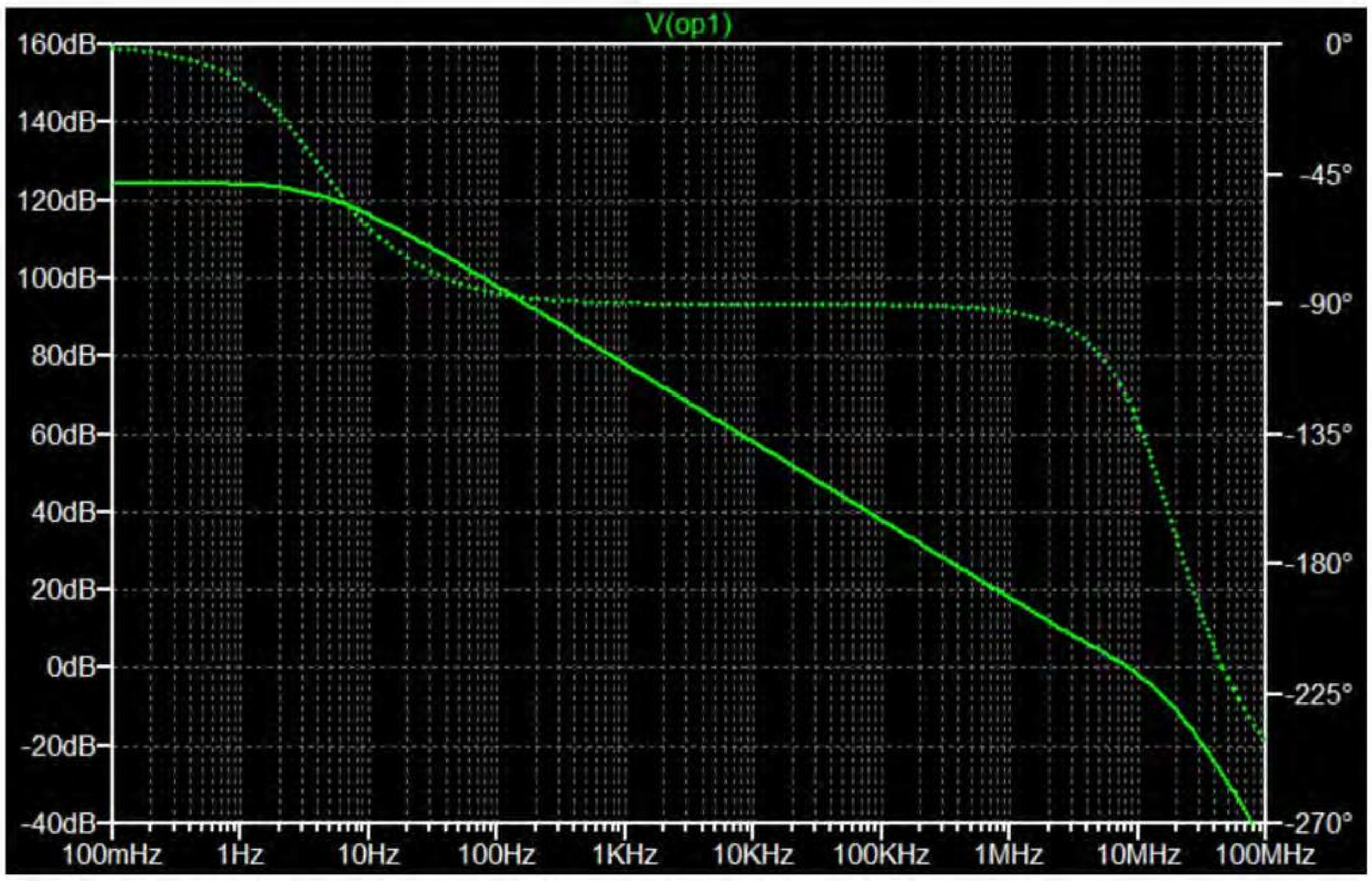


Fig.9: the open-loop response of the simulated OPAx134 op amp.

Compensation

We have discussed the fact that op amps are typically fully or partially compensated to provide stable operation in unity-gain or other low-gain configurations (eg, 3-5 times). This assumes that the capacitive loading or parasitics in the feedback loop are not sufficiently large to cause problems, which is not always the case in real circuits.

Assuming the problem cannot be fixed by layout changes to reduce parasitics (eg, increasing the separation between signal and ground tracks), the stability of an op amp circuit that is unstable or has unsatisfactory phase margin can be improved by circuit modifications. As in the design of the op amp devices, this process is called compensation. It works by adding zeros or poles to the loop gain.

The component values must be chosen correctly, because such modifications may not work or even make stability worse if not set up correctly. Also, changes in the load may shift the effect of any compensation circuit and hence affect stability, so working with a wider range of possible loads can be more challenging.

Isolation resistor compensation

A simple approach to stabilising an op amp circuit with a capacitive load is to insert a resistor between the op amp output and the load. This is often called an isolation resistor because it separates, or isolates, the load from the op amp's output. The technique is also referred to as an 'out of loop' compensation approach because the isolation resistor is not inside the feedback loop.

As this is a relatively straightforward approach, we will perform a simplified analysis of this circuit to obtain equations that can provide insights into the circuit's behaviour and help us select component values.

Fig.10 shows a version of the circuit in Fig.2 with the isolation resistor added to provide compensation (R_C). Like our analysis of the Fig.2 circuit last month, we will ignore the load resistance (R_L) to keep things simple.

Like last month, we will use the *s*-domain component values for the analysis. In Fig.10, we have a potential divider with R_O (=Z_O) forming the top part, and C_L and R_C in series forming the lower part (series impedance $R_C + 1 \div sC_L$ in the *s*-domain).

This potential divider will contribute to the overall loop gain. The loop gain without the load and compensation resistor, which is $-\beta A_O(s)$, is multiplied by the potential divider factor of:

$$\frac{R_C + 1/sC_L}{R_O + R_C + 1/sC_L}$$

As in previous examples, we can rearrange the equation to get polynomials in *s* in the numerator and denominator:

$$\frac{1 + sR_CC_L}{1 + s(R_O + R_C)C_L}$$

Like the uncompensated case, a pole is added to the loop gain, but with the pole frequency at:

$$f_p = \frac{1}{2\pi(R_O + R_C)C_L}$$

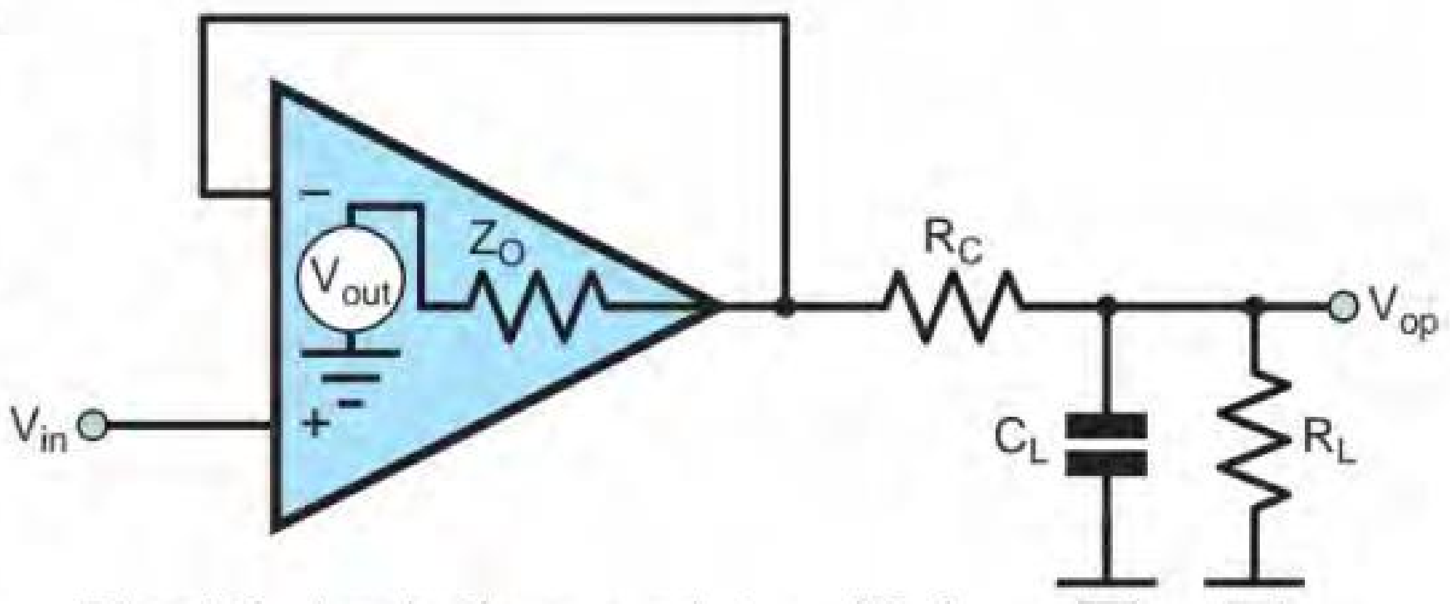


Fig.10: isolation resistor (R_C) compensation for improving stability.

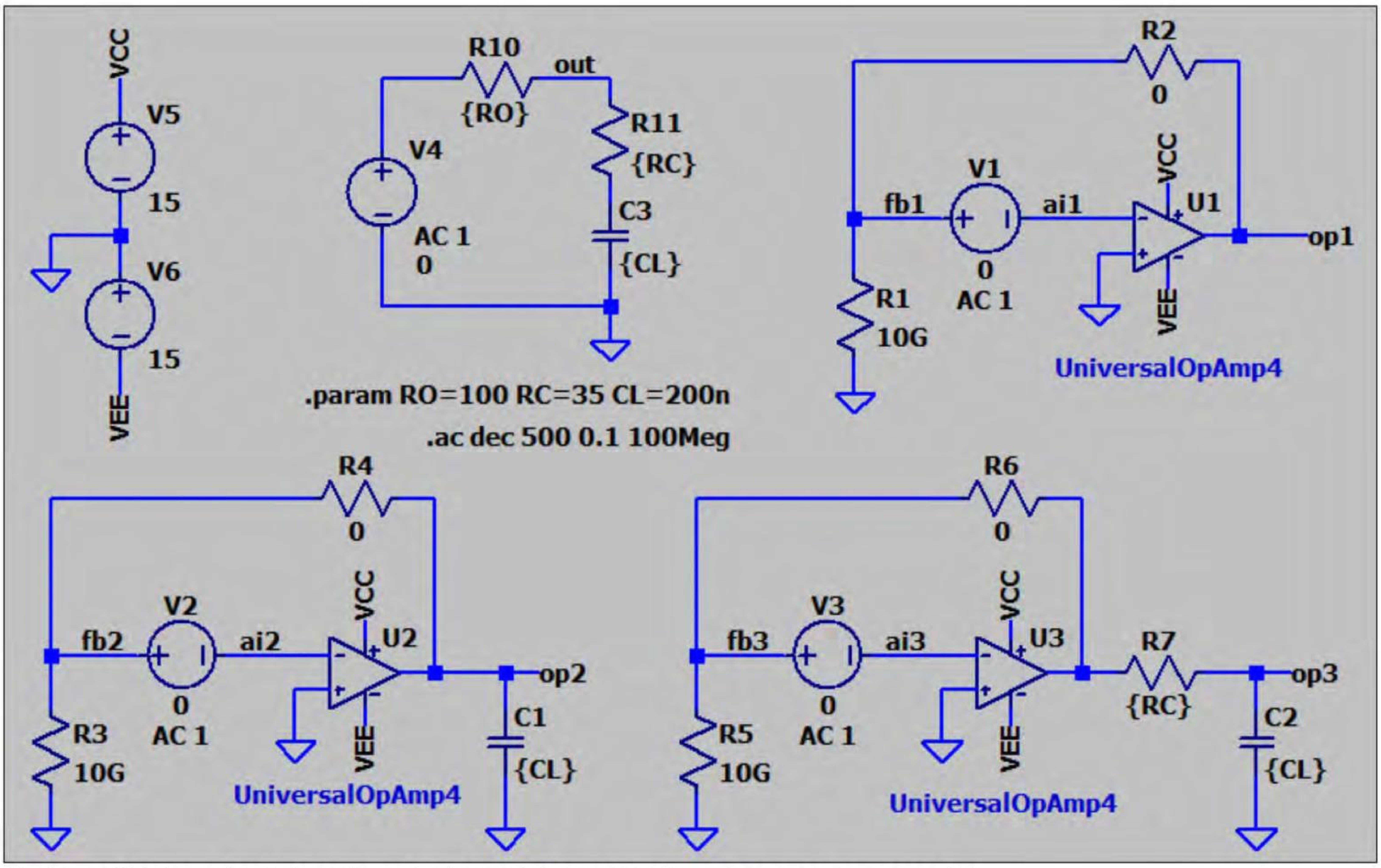


Fig.11: an LTspice schematic for investigating isolation resistor based op amp compensation.

That is different from the uncompensated case last month, which gave $f_p = 1 \div 2\pi R_O C_L$. The fact that this pole is at a lower frequency due to the resistance term increasing (from R_O to $R_O + R_C$) is not ideal, as on its own, it would make things worse. However, the key impact of the isolation resistor is to add a zero to the loop gain.

We have mainly been dealing with poles in our discussion of op amp circuits. Recall that the effect of a zero is opposite to that of a pole – it reduces the rate of decrease in gain (by 20dB/decade) and makes the phase shift more positive (eventually by 90°). A zero can compensate for the effect of a pole caused by the presence of a capacitive load or parasitics, as long as it is at a suitable frequency.

Zeros occur at frequencies where the *s*-domain gain equation is zero, that is, when the numerator of the equation is zero; in this case, $1 + sR_C C_L = 0$. As in previous examples, we substitute $s = j\omega = j2\pi f$ and find the magnitude, to get a zero frequency in Hertz:

$$f_z = \frac{1}{2\pi R_C C_L}$$

Compensation example

The LTspice circuit in Fig.11 can be used to explore the use of an isolation resistor for compensation. Like the circuit in Fig.3, this uses the UniversalOpAmp4 idealised op amp models configured for loop-gain simulation. The three op amp circuits are unloaded (U1), loaded (U2) and loaded with an isolation resistor (U3).

There is also an R_C circuit that includes the resistance and capacitance values relevant to loading and compensation. In this circuit, R10 represents the op amp's output impedance (R_C = 35Ω), R11 represents the isolation resistor used for compensation (R_C = 35Ω) and C3 represents the load capacitance (C_L = 200nF).

This circuit enables us to see the pole and zero due to the load and compensation separately from the op amp's feedback loop.

Investigating compensation components may need some trial and error, so in the Fig.11 circuit, the values are set via the LTspice parameter directive (`.param`), allowing all instances to be changed with a single edit. In LTspice, parameter values can be used in component values instead of directly using numerical values by writing the parameter name in curly braces (eg, `{RO}`) as the value.

The output resistance value of the op amps is entered as `{RO}` in the parameter table accessed by right-clicking the component.

With the values set, we can use the formulae discussed above to calculate the frequencies involved. We expect the pole in the loaded but uncompensated circuit to be at $f_p = 1 \div 2\pi R_O C_L = 1 \div (2\pi \times 100\Omega \times 200\text{nF}) = 7.96\text{kHz}$ (where 200nF = 200×10^{-9}F).

For the compensated circuit, we expect the pole to move to a lower frequency, $f_p = 1 \div 2\pi(R_O + R_C)C_L = 1 \div (2\pi \times [100\Omega + 35\Omega] \times 200\text{nF}) = 5.89\text{kHz}$.

We can expect the zero to be at $f_z = 1 \div 2\pi R_C C_L = 1 \div (2\pi \times 35\Omega \times 200\text{nF}) = 22.7\text{kHz}$.

5-year collections

2019-2023
£49.95

2018-2022
£49.95

2017-2021
£49.95

2016-2020
£44.95

Purchase and download at:
www.electronpublishing.com

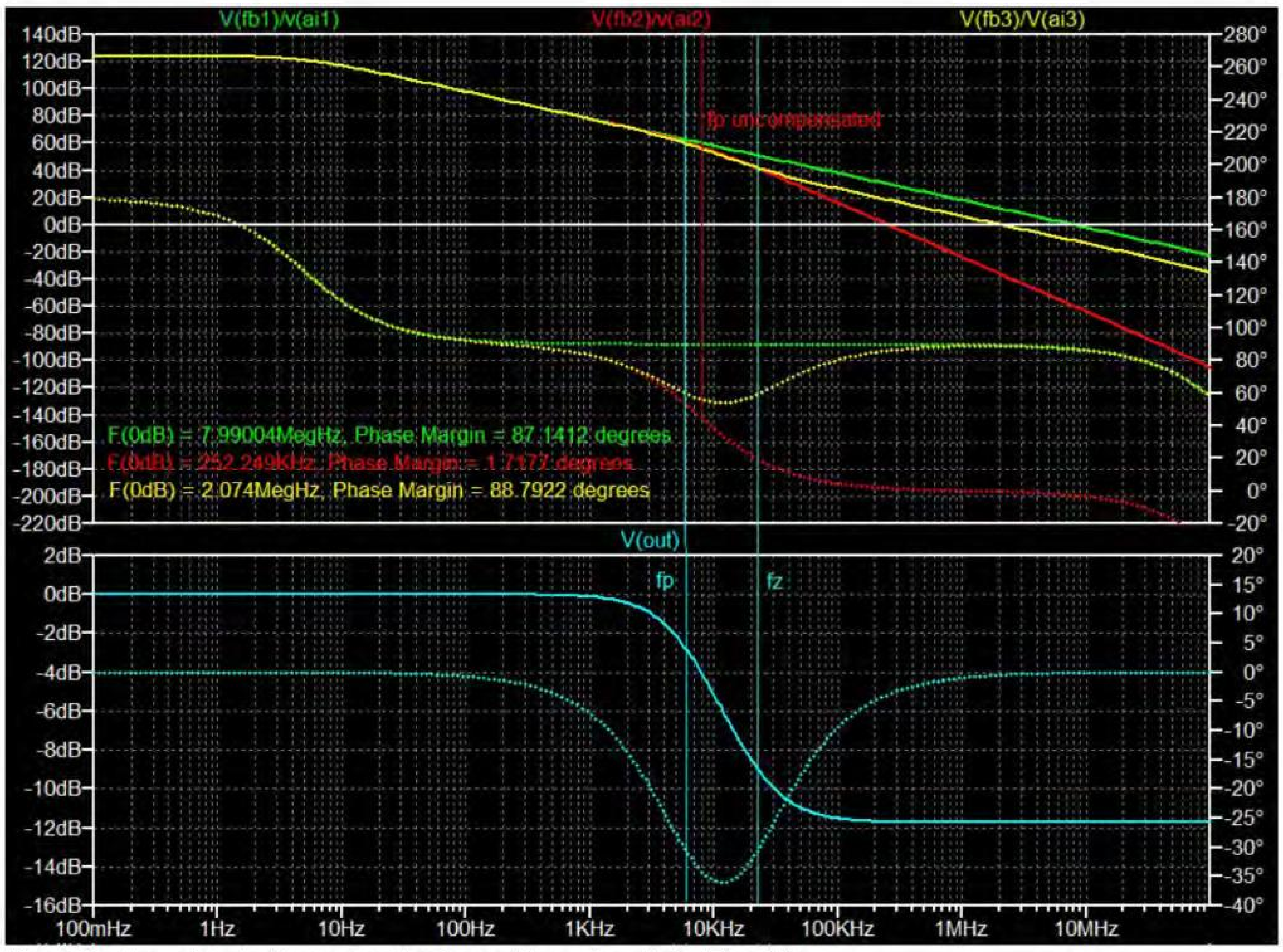


Fig.12: the simulation results from the circuit in Fig.11.

Simulation results

The simulation results are shown in Fig.12. The bottom pane shows the response of the RC model, while the upper pane shows the loop-gain responses of the three op amp circuits.

The RC model plot shows the effect of the pole causing a decrease in gain and an increasingly negative phase shift, which is reversed by the zero as frequency increases. The pole and zero frequencies (5.89kHz and 22.7kHz, as we calculated) are marked by the vertical cyan lines.

This plot is similar to the example showing a close pole and zero from our introduction to the topic in the January 2026 issue. The zero reverses the phase shift due to the pole before the full -90° is reached. It also limits the decrease in gain from the pole.

On the upper plot in Fig.12, the green trace shows the unloaded circuit response. This features a standard low-frequency op amp pole, in this case at about 5Hz (here phase has shifted -45° from the 180° due to negative feedback inversion at DC). The phase shift levels out at 90° above about 100Hz.

Phase margin annotation shows that it is very good for this circuit at 87°. The other circuit's responses follow the same curve at low frequencies.

There is also a high-frequency pole from the op amp beyond the upper range of the graph, which shows some effect on the phase shift of all the loop-gain plots above about 10MHz.

The red trace in Fig.12 shows the uncompensated, loaded op amp response. We can see the pole at 7.96kHz, which is marked with the vertical red line. This second pole, due to the load capacitance, occurs at a relatively low frequency, causes the phase shift to approach 0° well before the loop-gain reaches unity (0dB), resulting in the poor phase margin of 1.7° shown in the annotation.

This situation is similar to the circuits with poor stability from capacitive loading discussed last month.

The yellow trace in Fig.12 shows the compensated, loaded op amp response. This includes a pole and zero due to the combination of the load and isolation resistor. This pole and zero match those from the RC circuit model shown in the lower pane. The cyan vertical lines show their frequencies.

By comparing the yellow and cyan traces, we can see the effect of the compensated load on the loop-gain response. The phase margin annotation shows that the compensation has been effective at improving stability, with the value of 89° being similar to the unloaded circuit.

Adding responses

As mentioned above, the overall loop gain is equal to the basic loop gain of $-\beta A_O(s)$ ($\beta = 1$ in this case) multiplied by the potential divider factor due to the load capacitance, isolation resistor and op amp output impedance (modelled by the RC circuit).

Multiplying gain magnitudes of series stages in circuits translates to addition when we represent the gain in decibels due to the properties of logarithms: $\log(A \times B) = \log(A) + \log(B)$.

In the lower pane of Fig.12, we see the gain of the RC circuit change from 0dB to -12dB over the range of approximately 1kHz to 100kHz.

Looking at the loop-gain magnitude for the compensated circuit (yellow trace) in comparison with the unloaded circuit (green trace), which has a loop gain of $-\beta A_O(s)$, we see the gain of the compensated circuit drops by 12dB relative to the unloaded one over the same frequency range, then stays at that difference for higher frequencies.

For example, at 100MHz, the two gains are -23dB and -35dB.

Phase shift is related to delay, so contributions from circuits in series add. Again, we can compare the unloaded (dotted green) and compensated (dotted yellow) curves and see that if we add the RC model phase (dotted cyan) to the unloaded phase, we get the compensated phase curve.

The RC model has zero phase shift at both low and high frequencies due to the opposing action of the pole and zero. The RC model phase curve dips to -36° between the pole and zero frequencies, and the compensated circuit's phase curves dips by 36° from 90° down to 54° at the same frequencies.

At low frequencies, the curves for the loaded but uncompensated circuit (red traces) are initially similar to the compensated curve as both have poles at similar frequencies, However the compensation adds a zero that causes the compensated response to stop deviating further (gain), or bend back towards (phase) the unloaded case at higher frequencies.

Here, we see the presence of compensation mitigating or reversing the effects of the instability-inducing pole caused by the capacitive loading.

Isolation resistor value

If we know what the frequency of the zero introduced by the isolation resistor should be, we can use the formula above to calculate the resistor value. There is no single specific value that the frequency of the zero should be at for a given response to improve its stability, but there are some rules-of-thumb that can provide guidance.

Looking at the overall pattern of the response with the compensation in place, we can note the general fact that at a decade above a pole or zero frequency, the majority of its influence on phase has occurred, concluding that the zero introduced by the isolation resistor should be a decade or so below the unity-gain (0dB) frequency of the circuit that needs compensating to have the desired effect on phase margin.

With the help of the formula for the frequency of the zero, this rule-of-thumb can be used to help select an appropriate value. In Fig.12, we see that the condition is met. The unity gain (0dB) frequency of the loaded, uncompensated op amp loop-gain (red trace) is around 250kHz, which is a little over a decade above the zero frequency of 22.7kHz.

Alternatively, we note that about half of the phase shift influence occurs in the decade below the zero frequency, so if we set the frequency of the zero equal to the

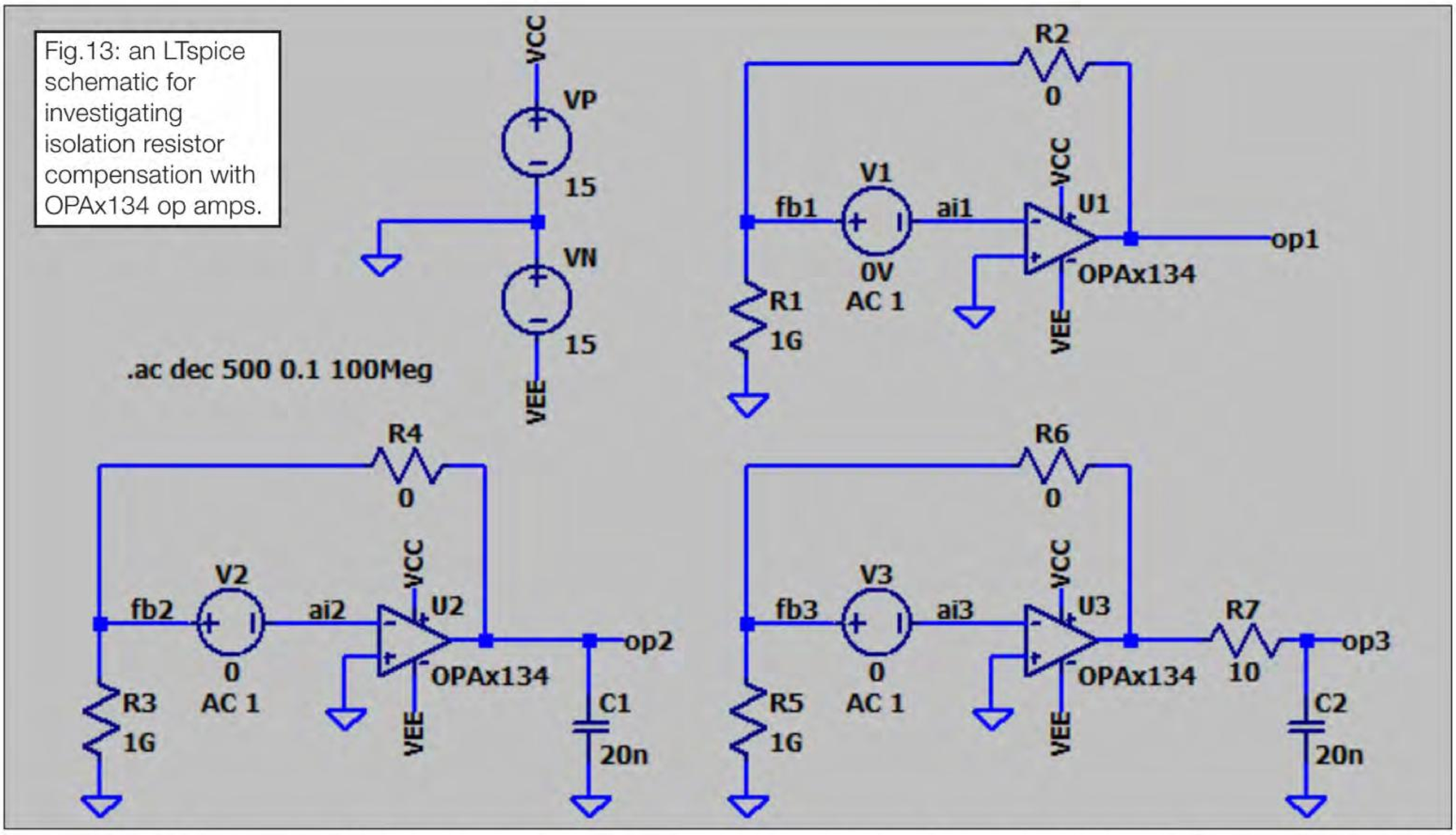


Fig.13: an LTspice schematic for investigating isolation resistor compensation with OPAx134 op amps.

unity-gain (0dB) frequency of the circuit that needs compensating, that may be sufficient to improve the stability of the circuit.

This will give a resistor ten times smaller than the previous rule; suitable values are likely to be between these two cases.

The pole due to the load capacitance would be fully cancelled if the zero was at the same frequency. This leads to another rule-of-thumb: that the frequency of the zero should be less than ten times the frequency of the uncompensated pole due to the load capacitor, so that the pole and zero are not too far apart.

Looking at the formulae for the pole and zero (above), we see they differ in the resistance terms; using R_C for the zero and ($R_C + R_O$) for the pole. If R_C was much larger than R_O, R_O would have little influence on the pole frequency, and the two would be very close. This implies the use of a large isolation resistance might be useful, but we have to consider the effect of load resistance.

So far we have ignored the possibility of a load resistance in parallel with the load capacitance (as shown in Fig.10). This was to keep the analysis simple and, in the simulations, to keep them matched to the analysis. However, a parallel resistance and capacitance combination is common (eg, a resistive load at the end of a long-shielded cable), so it must not be completely ignored.

We can perform an analysis similar to the one above with R_L included (as in Fig.10). The algebra involves more terms, so manipulating the equations is more cumbersome, but the results are similar; the pole shifts to a lower frequency and a zero is added to the loop gain.

The pole and zero expressions can be written in terms of the parallel combination of R_L with R_C for the zero, and with ($R_C + Z_O$) for the pole. They then have a similar form to the formulae above. This makes sense as 'looking' from the output into the circuit, the load resistance is in parallel with the combination of R_C and Z_O.

Also, if R_L is infinite (not present or ignored), the parallel resistance part of the expressions reverts to the simpler versions given earlier.

Including R_L has some impact on the choice of isolation resistor in terms of pole and zero frequencies, but a more important concern is that R_C and R_L form a potential divider at the op amp's output and hence change the overall closed-loop gain of the circuit. That is a major disadvantage of this approach to compensation for circuits requiring precision gain values.

In the case of the sound card circuit, the overall gain is adjusted as part of the setup process to get the right signal levels, and the measurement is performed relative to calibration, so this would not be a problem. Still, in other applications, an alternative approach to compensation may be needed.

OPAx134 example

Fig.13 shows an LTspice schematic for loop-gain simulation of OPAx134 unity-gain buffers that are unloaded (U1), loaded (U2) and loaded with isolation resistor compensation (U3). The results are shown in Fig.14. The unloaded and compensated circuits are stable; with similar phase margins, the loaded circuit is unstable.

The uncompensated circuit has a unity gain frequency of 1.085MHz (see the annotation in Fig.14). Using the first

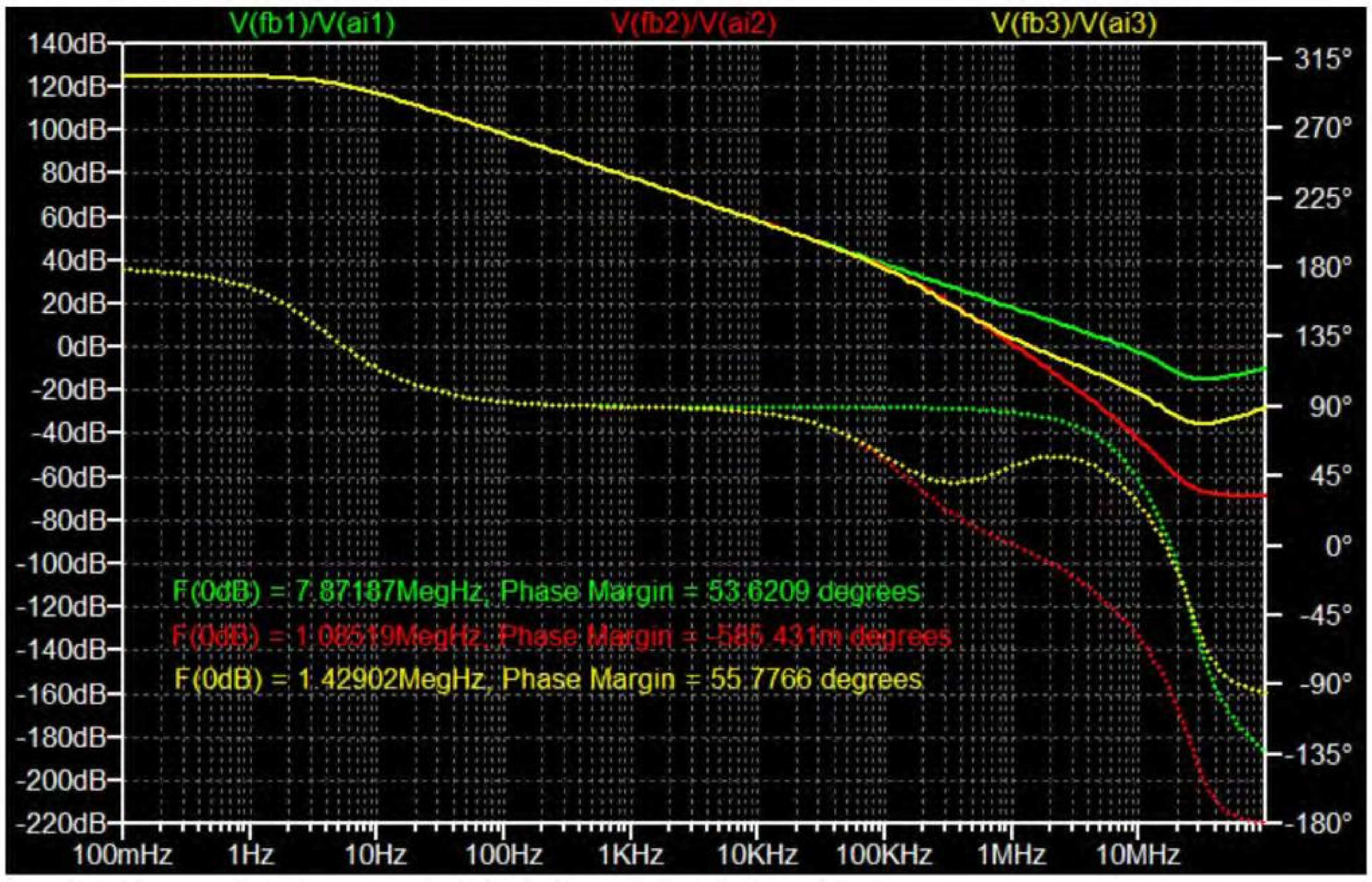


Fig.14: the results from simulating the circuit in Fig.13.

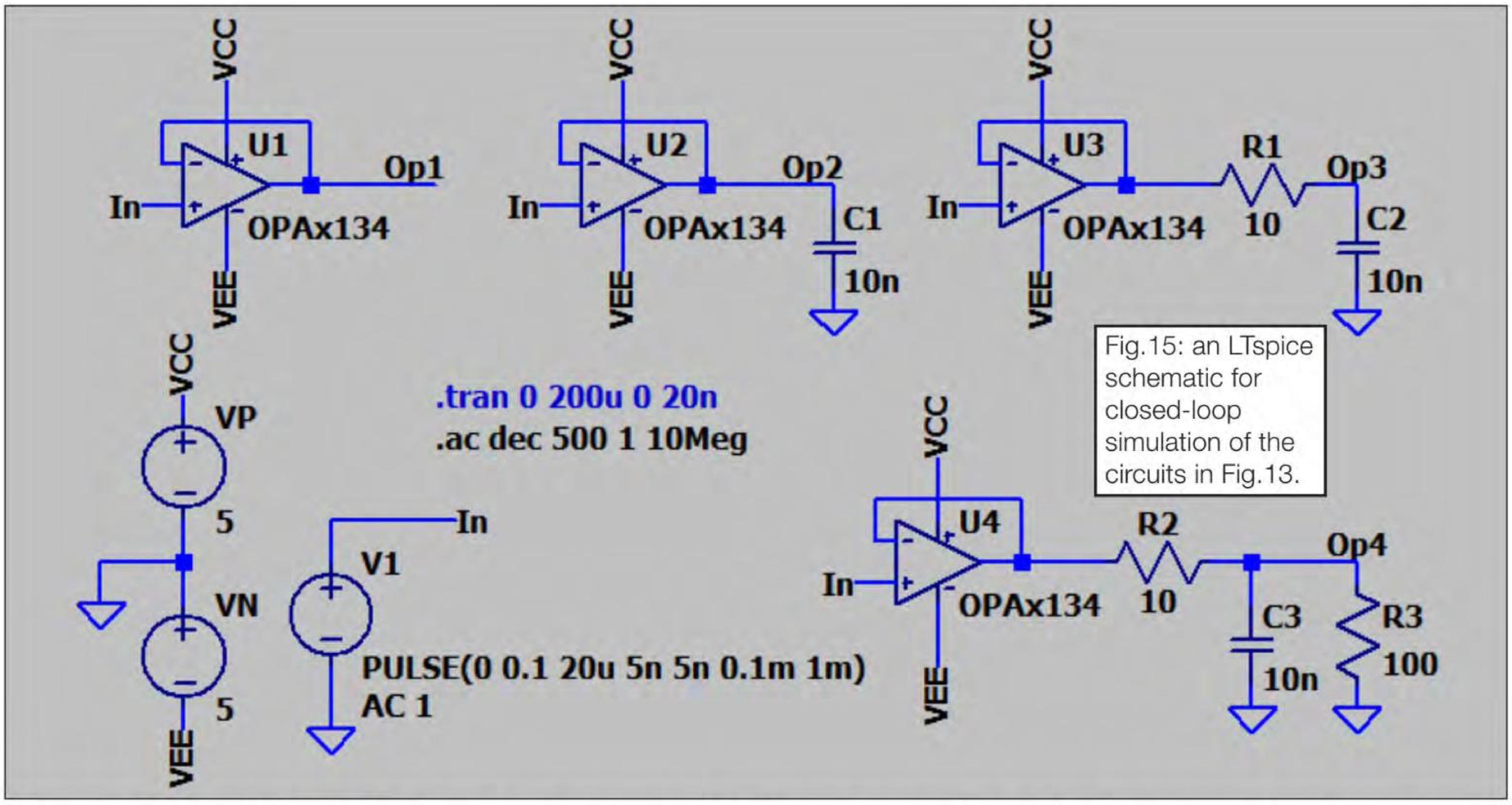


Fig.15: an LTspice schematic for closed-loop simulation of the circuits in Fig.13.

unity-gain-frequency-based rule above, we can get a zero at this frequency using $R_C = 1 \div 2\pi f_p C_L = 1 \div (2\pi \times 1.085\text{MHz} \times 20\text{pF}) = 7.3\Omega$ ($20\text{pF} = 20\times10^{-12}\text{F}$). Using a decade lower frequency for the pole, in the variant of the rule, would give 73Ω.

The pole frequency due to the uncompensated load capacitor can be calculated if we know the op amp's output resistance. From Fig.8, we can assume it is about 54Ω, as the curve is quite flat around this value over the relevant range. This gives $f_p = 1 \div 2\pi R_O C_L = 1 \div (2\pi \times 54\Omega \times 20\text{pF}) = 146\text{kHz}$. Using the pole-based rule above, we get a maximum zero frequency of 1.46MHz.

Using the same calculation as in the previous paragraph, the minimum isolation resistor value is 5.5Ω, which is not far from the 7.3Ω obtained via the other rule.

The results in Fig.14 were obtained using a 10Ω isolation resistor, a round value a little larger than the minimum values from above. The shapes of the curves in Fig.14 follow a similar pattern to Fig.12, but with some more complexity due to the op amp's high frequency behaviour.

Fig.15 shows an LTspice schematic for simulating the closed-loop behaviour of the OPAx134 buffer unloaded (U1), loaded (U2), compensated (U3) and with a resistive load added (U4). Fig.16 shows the AC analysis results, and Fig.17 the transient analysis with a pulse input.

In Fig.16, the gain response of the loaded circuit (red trace) shows a large peak typical of an unstable circuit. The two compensated (and stable) circuits have much smaller peaks. The circuit with the resistive load (cyan trace) has a gain of -0.83dB (0.909) rather than the unity gain of the buffer without the resistive load (yellow trace) due to the potential divider effect discussed above.

In Fig.17, the unloaded (green trace), compensated (yellow trace) and compensated with resistive added load (cyan trace) circuits all have similar overshoot and minimal oscillation (stable behaviour). The unloaded and compensated circuits have similar phase margins, so the overshoot is similar, as discussed last month.

The reduced gain of the circuit with the resistive load is also seen here.

Next month, we will look at alternative approaches to stabilising an unstable or marginally stable op amp circuits.

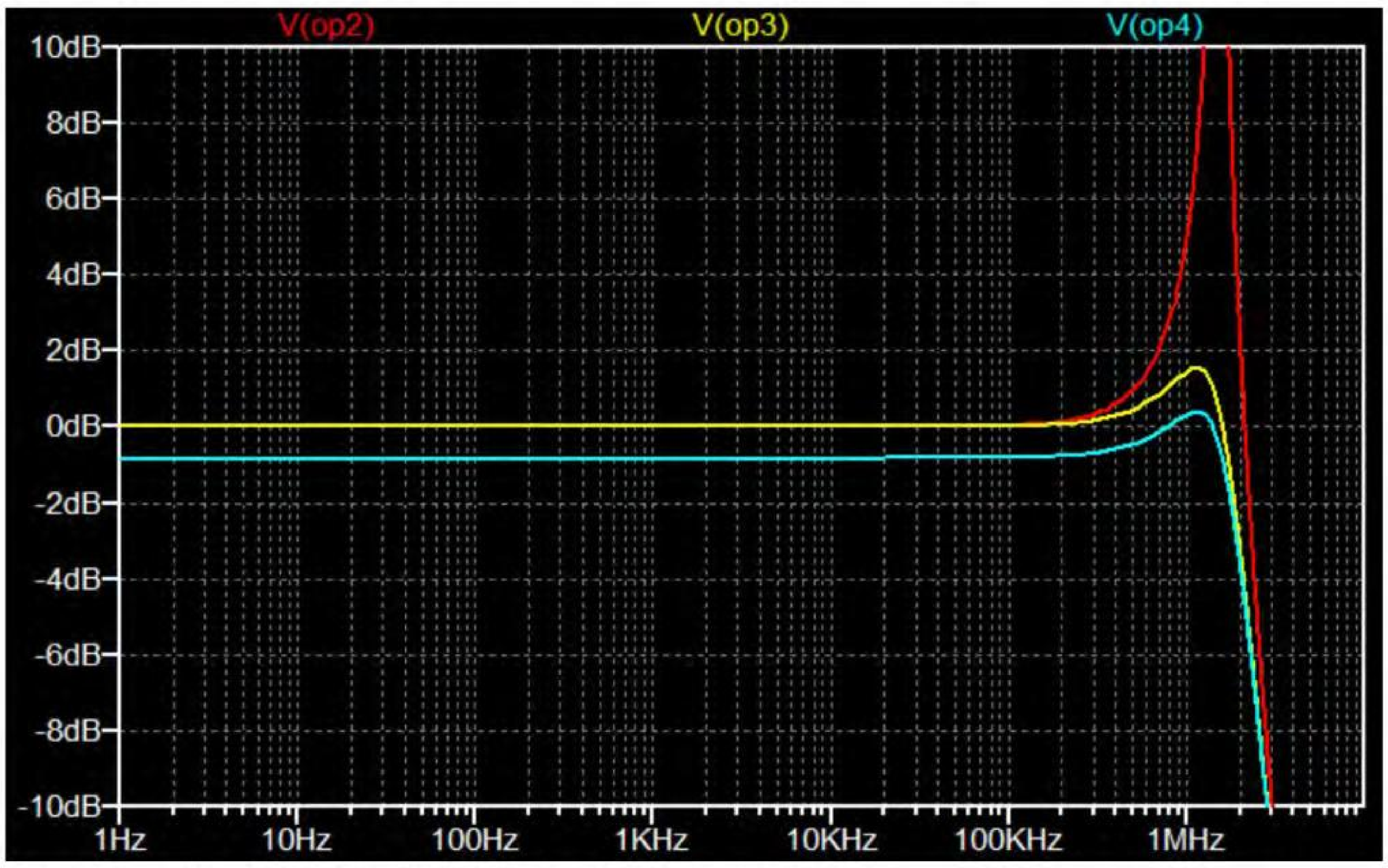


Fig.16: results from the AC analysis simulation in Fig.15.

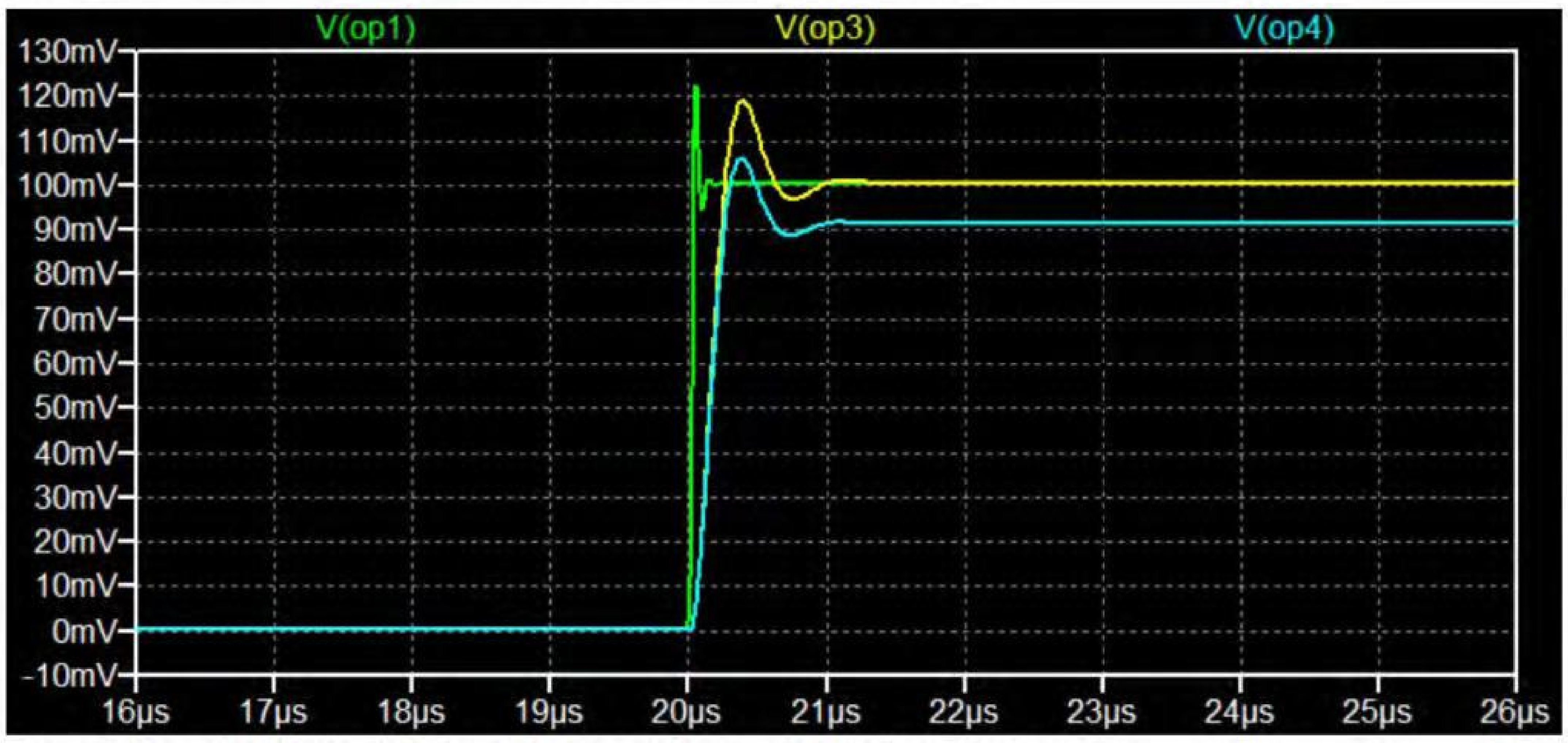


Fig.17: results from the transient analysis simulation in Fig.15.

AOShop

The home for specialist audio, analogue and historic components – provided by Jake Rothman, *PE's Audio Out* columnist. The AOShop is your best bet for classic analogue 'NOS' (new old stock) components, including all parts for *Audio Out* projects and designs.

Ge semiconductors

Small-signal PNP transistors	
NKT214F, OC57, OC59, 2N1377, 2N525	**£1.00**
Low-noise PNP transistors	
GET106	**£2.50**
Small power PNP transistors	
AC153, AC153K, AC188	**£1.50**
Small power NPN transistors	
AC176, AC141K	**£2.00**
High-voltage PNP transistors	
OC77, CV7001	**£1.50**
TO3 PNP power transistors	
OC22, CV7054 (OC23), OC25, OC35, OC36, AD143, AD149, AD161, AD162	**£2.00**
AD140	**£3.50**
AD149, AD161/2 matched pair	**£5.00**
RF PNP transistors	
OC41, OC42, 2SA12, 2SA53,AF124, AF178, GET872A	**£1.50**
NPN transistors	
OC139, OC140, ASY73	**£2.00**
Diodes	
CV7049 (OA10), CG92 (OA91)	**£0.50**

Si semiconductors

Diodes	
ZC5800 RF Schottky	**£0.20**
Low-noise silicon transistors	
BFW16A, 2SC3071, 2SC3068, 2SA1016K, 2SC2362K, 2SA970BL	**£1.50**
2SC2204, 2SD655, BC550C	**£0.50**
ZTX651	**£0.30**
ZTX751	**£0.50**
RF transistors (suitable for Theremin)	
BF199	**£0.50**
Audio power MOSFETs	
Exicon 10N20, 10P20	**£6.50**
Hitachi 2SJ99, 2SK343	**£3.50**
Hitachi 2SJ56, 2SK176	**£8.50**
JFETs	
BFW11, BFW10, TIS73L, J177, J113, U1994, U1898, 2SJ176, J201	**£1.00**
J175, J176, J112, J111, 2N3820, 2N5457, BF244, 2N5460, J230	**£0.60**
Small power output/driver transistors	
2SB649A, D669A, 2SA1208, 2SC2910, MJE253G, MJE243G, 2SA1725, 2SC4511	**£1.20**
BD139,BD140, BD135, BD136, BD435, BD436, MPSA63, BCV46	**£0.50**
MOSFETs	
ZVP2106A	**£0.30**
Dual-gate 3SK45, BFS28	**£1.50**
Metal-cased transistors	
BCY71	**£0.30**
Amplifiers	
LM318 high-speed op amp	**£0.35**
µA709 metal-cased op amp	**£2.00**
5534H metal-cased op amp	**£3.00**
LM384, TDA2030A, TDA2050V	**£1.50**
TAA435 (Mullard power amp driver for AD161/AD162)	**£1.20**

Synthesiser ICs

That/dbx 2180 VCA/VCF	**£6.00**
CA3080 VCA/VCF (vocoder)	**£2.80**
CA3280 VCA/VCF	**£4.00**
LM13600/ LM13700 VCA/VCF	**£2.00**
CA3086 transistor array	**£1.00**
PT2399 echo/delay	**£1.50**

Dual transistors

2N2639, 2N2223, 2N2910 (NPN)	**£4.00**
E401 (JFET Moog)	**£4.00**
2SK2145-Y dual JFET	**£0.80**
2N5564 JFET	**£8.00**
HN3C51F, HN3A51F	**£1.00**
DMMT3904/6, HN1A01F, HN1C01F	**£0.50**

Loudspeakers

PE Mini-Monitor	
Volt PE165 6.5-inch woofer (each)	**£85**
Morel MDT29 tweeter (each)	**£25**
Kit *pair* of PE165/MDT29 plus Volt crossover parts and PCBs	**£299**
Monacor DT-28N tweeter (each)	**£35**
Vifa 19mm BC20SC15-04 tweeter (each)	**£15**
Volt crossover inductors	
1.2mH, 1.5mH, 2mH, 2.7mH, 0.5mH (tapped at 0.3mH) (each)	**£5.85**
Fully assembled and tested high-quality speaker prototypes – ask for details	
LS3/5As and other similar speaker systems (pair)	**£200-£350**
Fully tested reclaimed speakers	
Vifa BC14 5-inch woofer (each)	**£10**
Vifa TC26 1-inch tweeter (each)	**£10**
Low-price speakers	
Philips 4-inch 4070	**£2.00**
EMI 10x6-inch, 30Ω Alnico	**£7**
64mm 64Ω neodymium	**£1.20**

Capacitors

Note '10/63' denotes '10µF 63V'.

Polyester	
3.3/100, 4.7/250, 4.7/63	**£1.00**
5.6/63, 8.2/63, 10/63	**£2.00**
Mullard 'Mustard' C296 0.22/400	**£2.00**
Polycarbonate	
Axial 2.2/63 1%, 4.7/160, 6.8/63	**£1.00**
Radial	
6.8/160V, 10/63	**£2.00**
Reclaimed 22/63	**£2.00**
Polystyrene	
Philips 1% 4.7nF/160, 6.2nF/500, 12nF/63, 22nF/63, 110nF/63, 24nF, 2nF	**£1.00**
RIFA 1% 100nF/100, Suflex 90.9nF 0.5%	**£2.00**
Suflex 2.5% 10nF/63 (rad. or ax.)	**£0.50**
Radiation resistant	
Siemens cellulose acetate MKL 2.2/25	**£0.80**
Electrolytic – Mullard blue 017 series	
10/25, 22/25, 100/10	**£0.50**
150/40, 470/40, 1000/40	**£1.00**
Tantalum – axial metal cased	
22/50, 47/35, 68/25, 100/20, 120/10, 150/16, 220/10, 330/6	**£2.00**
22/35, 33/35, 47/20, 68/15, 100/10, 150/6	**£1.25**
4.7/50, 6.8/35, 10/25, 10/35, 22/15	**£1.00**
Axial moulded-case tantalum	
Kemet axial 6.8/10	**£0.30**
Kemet radial 33/10	**£0.40**
STC radial 100/20	**£1.50**
Tantalum bead	
22/50	**£1.00**
470/3	**£2.00**
680/6.3	**£3.00**
Wet tantalum	
220/25 axial	**£3.20**
Castanet button 140/30, 470/3	**£3.20**
Hughes 540/10	**£3.20**
Bipolar	
Hermetic bipolar tantalum 16/35	**£3.20**
Elcap axial 10/50	**£0.50**
Generic radial 100/16, 470/35, 100/ 63, 22/35, 4.7/35, 220/16	**£0.50**
Philips solid-aluminium (axial) 121/123	
47/16	**£1.00**
330/6.3	**£2.00**
100/35	**£4.00**
Philips Pearl 122 series (radial)	
10/16	**£0.30**
Silvered mica (radial)	
1nF/500 1%	**£1.00**
Trimmer capacitors	
Vishay plastic-film 4-40pF	**£0.80**
Vishay plastic-film 5.5-45pF	**£1.00**
Vishay plastic-film 5-80pF	**£1.20**
Vishay plastic-film 10-250pF	**£2.00**
Mica 1-12pF, 2-40pF	**£1.00**

Audio transformers and inductors

Eagle transformers	
LT44, LT722 driver, LT700, LT723 500Ω output	**£2.50**
LT30 500mW output	**£3.50**
5:1 interstage (fuzz box)	**£1.50**
Repanco	
T/T3 splitter transformer	**£4.00**
CH2 5mH RFC	**£2.00**
Balanced output transformer	
Vigortronix 600Ω VTX-101-007	**£10**
Vigortronix 600Ω VTX-101-3001	**£10**
Vigortronix 600Ω VTX-101-3002	**£15**
Gardners 150Ω	**£10**
Reclaimed BBC LL74/MPC nickel core 600Ω	**£12**
Low-level 1:1 600Ω (modem)	**£2**
Reclaimed mic input transformer	**£15**
Inductors	
82µH, 4.7mH, 100µH, 270µH, 10µH, 14µH (low Z)	**£0.50**
4-inch ferrite rod with MW and LW windings	**£3.00**

Special resistors

Bourns wire-wound trimmer 5kΩ 3059 JM panel-mount	**£2.00**
Thermistor RA53, R13	**£4.00**
A13	**£2.00**
Thermistor CZ1, CZ6	**£1.50**
Holco H2 2.2MΩ 1W, 1%	**£1.00**
Welwyn 1GΩ 2W	**£1.00**
5k Bourns 3321H cermet trimmer	**£0.50**

Potentiometers

Bourns 81	
25kΩ lin cermet	**£2.00**
25kΩ lin, 5kΩ lin cond. plastic	**£3.00**
Bourns 91	
10k dual-gang lin or log	**£10.00**
Plessey moulded-track	
5kΩ log with switch	**£3.00**
50kΩ A/log	**£3.00**
Mil 250kΩ lin dual	**£5.00**
Alpha	
16mm 4.7kΩ A/log	**£0.80**
220kΩ A/log	**£0.80**
10kΩ lin centre-detent dual-gang	**£1.50**
Allen Bradley J series/Honeywell	
10kΩ lin 1 million cycles	**£5.00**
Blore Edwards AB 45 dual 5kΩ A/log with switch	**£3.50**
Alps RK9 dual-gang 5kΩ RD law	**£4.00**
Alps 50k log tapped motor. stereo	**£5.00**
BI P260 500kΩ log conductive plastic 1 million cycles	**£2.00**
Colvern wire-wound 100kΩ or 50kΩ dual-gang 3W	**£5.00**
Mil Spec hermetic 10Ω	**£8.00**

Miscellaneous

PCBs	
Silent Stomp Switch (pair)	**£4.00**
Penta-fuzz Bender (*PE*, Feb26)	**£3.00**
Pre-driver (*PE*, Jan26)	**£3.00**
Pre-driver jack bd (*PE*, Jan26)	**£1.50**
Fuzz Box (*PE*, Nov25)	**£3.00**
Fuzz Box jack bd (*PE*, Dec25)	**£1.50**
Dual JFET Buffer (*PE*, Oct25)	**£2.00**
Mosfet Buffer (*PE*, Oct 2025)	**£2.00**
Dual Mosfet Buffer (*PE*, Oct25)	**£2.00**
Cascode Amp (*PE*, Sep 2025)	**£2.00**
Cascode PS (*PE*, Sep 2025)	**£2.00**
Pocket Theremin (*EPE*, 1996)	**£2.00**
Elysian Theremin (*EPE*, 1996)	**£6.00**
Synth VCF, VCO (*EPE*, 2017)	**£3.00**
48V PSU (*EPE*, 2019)	**£3.00**
SMT dual transistor adapter	**£0.40**
Relays	
STC hermetic 4190 24V DPDT	**£4.00**
Switches Blore Edwards/Plessey	
MH1 4-pole 3-way	**£3.00**
Elma 01 2-pole 6-way	**£6.00**
3PDT guitar pedal stomp switch	**£2.50**

Contact

Jake Rothman
Manchester House
Market Street, Craven Arms
Shropshire SY7 9NN
(visit by appointment)
+44 (0)1597 829102
jrothman1962@gmail.com

Minimum order £5.00 inc post

Payment

PayPal, bank transfer, cheques (payable to "J Rothman", UK pounds only). No VAT payable

Postage

Small Jiffy bag £2.99
Small package £4.99
Big boxes and overseas at cost – ask for a quote

Where did it go?

Techno Talk

From visible mechanisms and fixable machines to sealed laptops, microscopic sensors and optical MEMS. Join me on a ramble through modern technology as it grows more powerful and more mindboggling.

Max the Magnificent

As usual, my poor old noggin is full of random thoughts, ricocheting furiously around my cranium like they've missed the last train home.

I recall once reading that the average person a century ago could understand and repair almost everything in their home. Not perfectly, perhaps, but well enough.

Household lighting required routine maintenance: wicks trimmed, chimneys cleaned, flames adjusted. Shoes were re-soled, not replaced. Chairs were tightened, clocks were oiled, bicycles were stripped and rebuilt on kitchen tables. Early motorcars demanded mechanical sympathy from their owners. Things were expected to wear, to fail, and—crucially—to be fixed.

Today, many of our most important devices arrive as immaculate, sealed monoliths that are astonishingly capable, yet fundamentally uninterested in being understood or maintained. "If it breaks, throw it away and get a new one" seems to be the sentiment of the day.

Where did all the screws go?

This line of thinking led me to a broader observation: from one perspective, technology appears to grow simpler with time. One might argue that early technologies "externalised complexity." A hundred years ago, a mechanical typewriter was a forest of levers, gears, and linkages. It flaunted everything you needed to see, hear, and touch to understand how each character appeared on paper.

If we fast-forward to today's desktop, an inkjet printer (for example) conceals its true complexity behind its plastic shell. The tangible mechanisms of yesteryear have been replaced by silicon chips, firmware and tiny stepper motors that rarely reveal their secrets.

Even the humble print head, which looks deceptively simple, is a marvel of micro-engineering, rapidly heating microscopic chambers, vaporising ink and hurling it at the paper as a constellation of dots so small as to be indistinguishable to the human eye; all guided by algorithms we never see and physics most of us never consider.

But that's only half the story. Over time, complexity doesn't just become hidden; it ends up being sealed off. In the first phase, complexity is visible and exposed.

In the second phase, it's encapsulated under layers of abstraction and clever design.

In the third, it becomes almost inaccessible to the hobbyist, the maker, and even the trained technician: a single, integrated entity that you cannot easily repair, diagnose, upgrade, or even look inside. Modern laptops are perhaps an archetypal example. They're sleek, powerful, and almost hermetically sealed, leaving only a shiny case sporting a keyboard and a glowing screen.

When fixable tech feels radical

There's always "the exception that proves the rule", as they say. In this case, the exception is a company called Framework Computer (**https://frame.work/**), which is on a mission to create products designed from the ground up to last longer, be fixable and be upgradeable.

Enter Framework's Laptop 12 (**https://frame.work/laptop12**), available either as a fully assembled system or as a build-it-yourself kit. This bodacious beauty combines contemporary performance with old-fashioned openness.

You can configure it with the latest technologies, including Wi-Fi 6E, up to 48GB of high-speed DDR5 memory and up to 2TB of solid-state storage, tailoring the machine precisely to your needs.

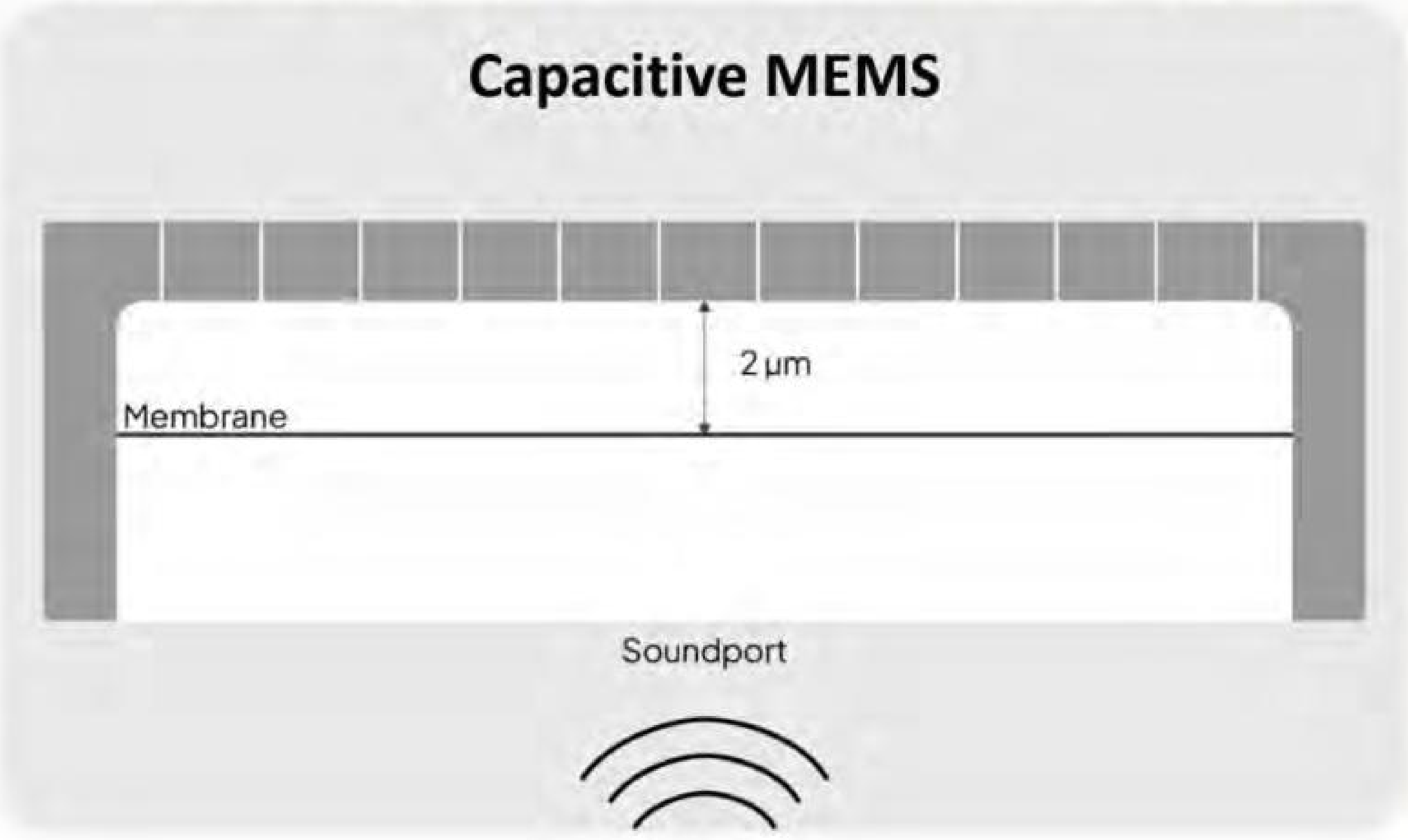
Its shock-absorbing metal chassis even meets MIL-STD-810 durability standards, helping it survive the inevitable bumps and drops of everyday life. Best of all, the Laptop 12 is explicitly designed to be easy to repair, upgrade, and maintain, giving it the potential for a far longer useful life than most modern laptops.

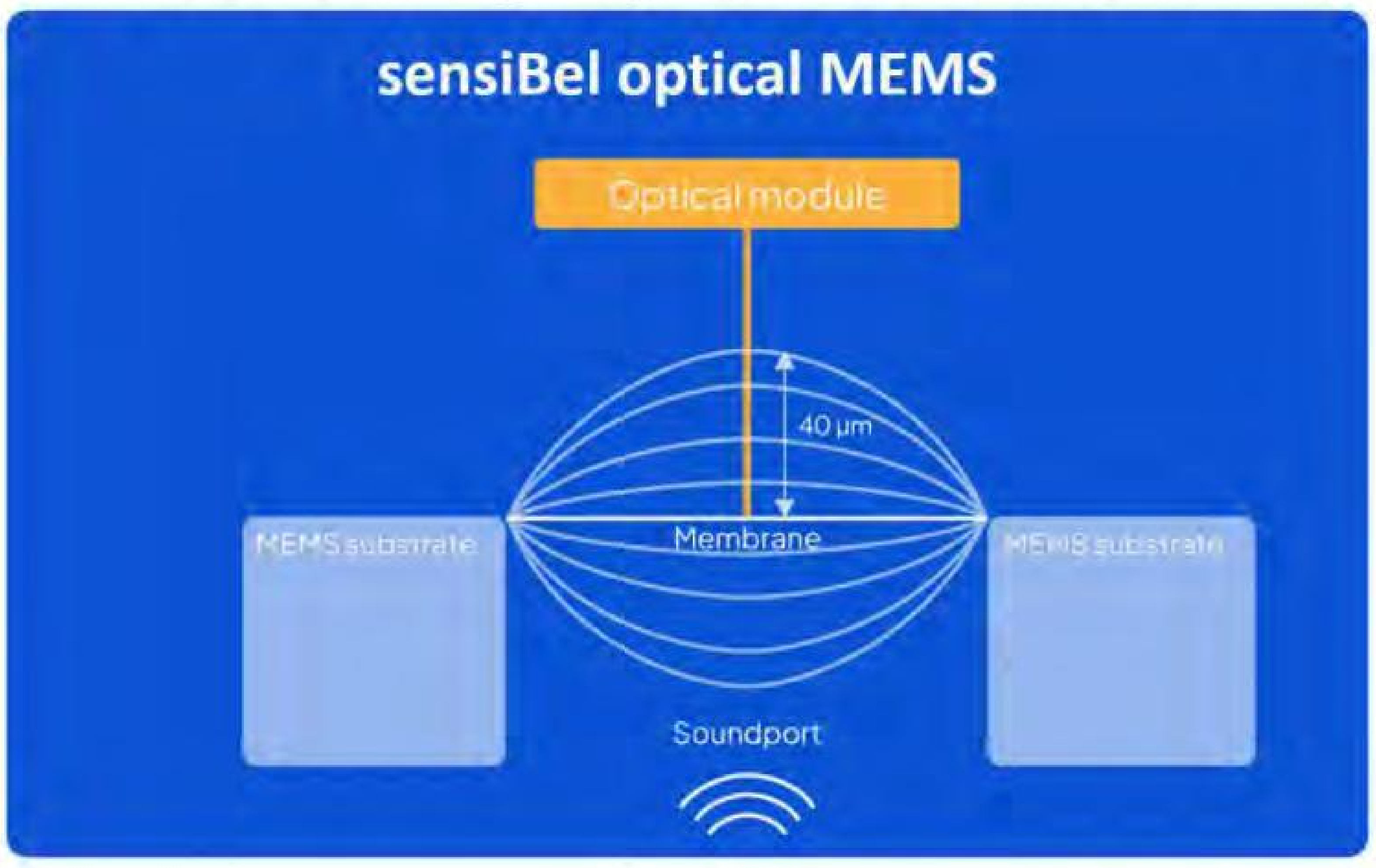
That sounds super!

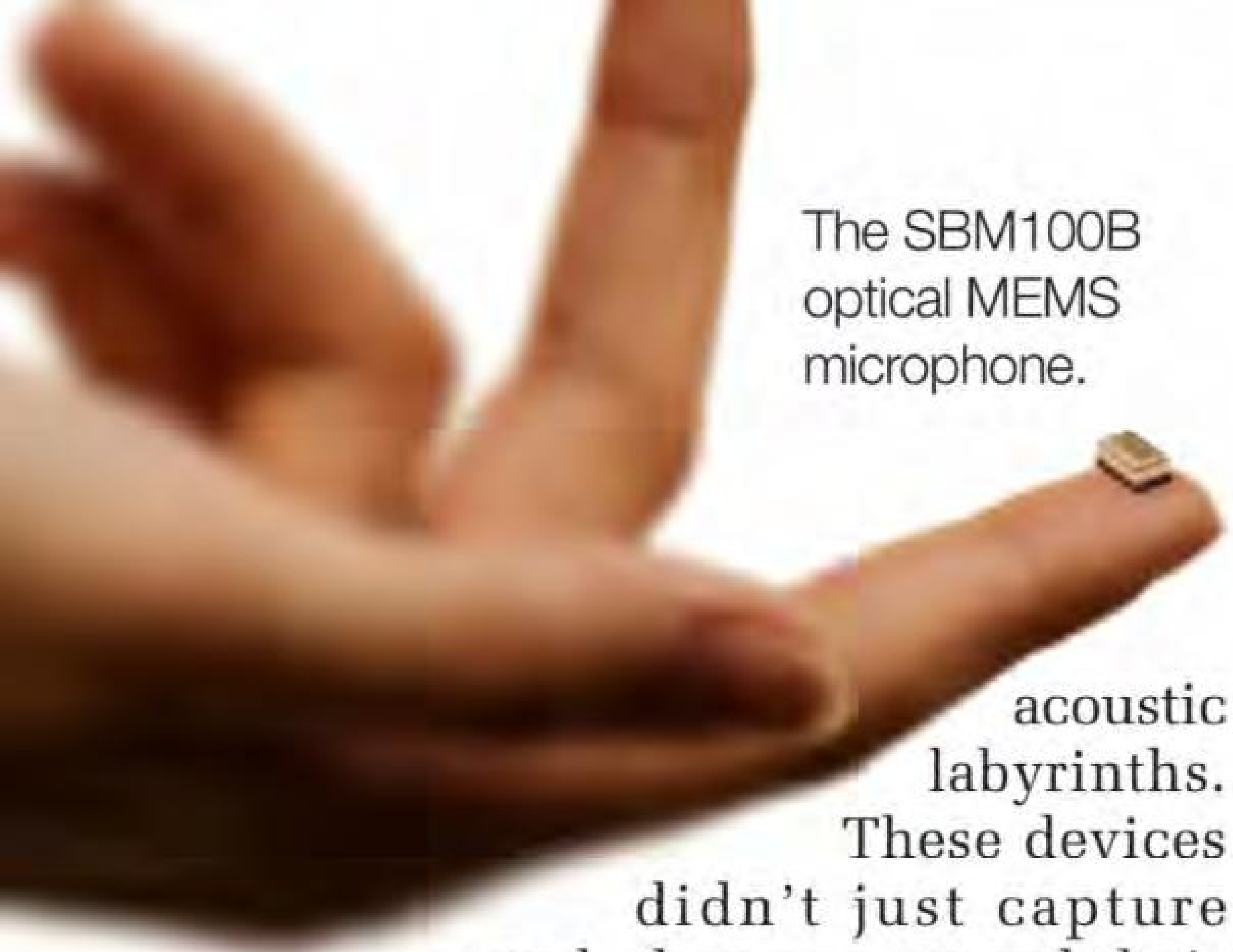
Another hallmark of modern technology is that it doesn't just retreat behind enclosures—it also tends to shrink. With a few notable exceptions (TVs spring immediately to mind, because bigger really is better there), many of the technologies we rely on every day have been on a relentless diet. Functions that once demanded bulky hardware now fit into spaces so small that they border on the invisible.

Take microphones, for example. Classic studio microphones from the 1920s to the 1950s were magnificent monsters: heavy, imposing, and impossible to ignore. While some designs leaned toward a more Bauhaus-like functionalism, many could qualify as works of art, strongly echoing the Art Deco movement.

Some were so large that you could barely see the user behind them; their polished grilles concealed intricate mechanical structures, delicate diaphragms, coils, transformers and


Capacitive MEMS


sensiBel optical MEMS


The SBM100B optical MEMS microphone.

acoustic labyrinths. These devices didn't just capture sound; they announced their presence (no pun intended), both visually and physically.

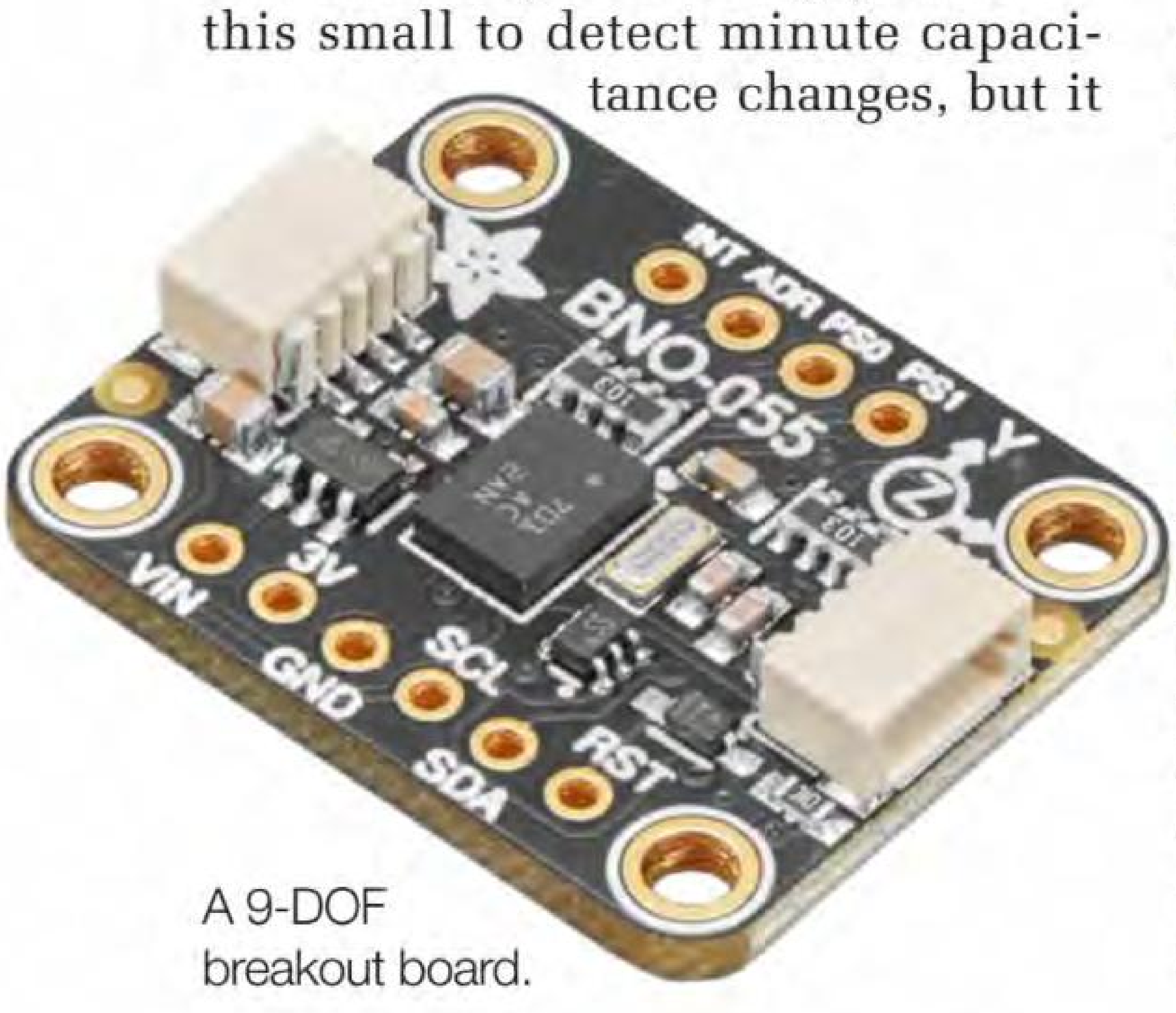
Fast forward to today, and many microphone-related tasks are handled by MEMS capacitive microphones that are scarcely larger than a grain of rice. Fabricated using silicon processes more commonly associated with integrated circuits, these tiny marvels rely on microscopic diaphragms and capacitive-sensing structures etched directly into silicon.

They consume mere microwatts, deliver impressive dynamic range, and can be embedded almost anywhere: smartphones, earbuds, laptops, smart speakers, toys and even cat collars. Once again, complexity hasn't vanished; it's simply been compressed, hidden, and mass-produced at a scale that would have seemed like science fiction to the engineers who built the classic studio giants of yesteryear.

It can be difficult to wrap one's head around the incredibly small structures we're talking about here. In the case of capacitive MEMS microphones, for example, the sound port (the hole where the sound comes in) is typically only about 1mm in diameter.

One plate of the capacitor is formed by a moving membrane (diaphragm), while the other is a rigid backplate. That backplate must be perforated with microscopic holes to let air pass through so the membrane can move, but those holes are a significant source of noise.

Another limiting factor in such microphones is the tiny gap, typically around 2μm, between the membrane and the backplate. The gap must be this small to detect minute capacitance changes, but it restricts membrane movement, limiting dynamic range and increasing the risk of clipping.



A 9-DOF breakout board.

The reason I mention this here is because I recently had an interesting chat with the guys and gals at sensiBel (www.sensibel.com). They have introduced the world's first optical MEMS microphone, the SBM100B.

Crucially, this new device dispenses with the backplate entirely and therefore with the noise-inducing holes that go with it. This also allows the membrane to move a remarkable ±40μm. Membrane motion is detected optically using a laser, and the entire packaged device is still small enough to sit on the tip of your finger.

Meet the SBM100B

The result is a high-fidelity, studio-quality microphone capable of handling sound signals 250 times stronger than those of conventional capacitive MEMS designs, while also delivering a noise floor roughly five times lower. In short, this is the sort of microphone that can finally bridge the gap between studio-grade audio and truly miniature form factors.

How low can we go?

A classic example of a shrinking technology is the gyroscope. When I was a lad, the three-axis gyroscopes used in early-1960s bombers were formidable electromechanical instruments. They were built around substantial spinning rotors running in evacuated housings, mounted within a heavily engineered inertial platform.

These units could approach the size of a small oil drum and cost hundreds of thousands of pounds in today's money.

By comparison, these days you can get a MEMS device that packs a 3-axis gyroscope, along with a 3-axis accelerometer, a 3-axis magnetometer, and a 32-bit microcontroller to perform sensor fusion, in a package only a few millimetres wide. Such a device can be obtained on a postage-stamp-sized breakout board at a price affordable even to humble hobbyists (www.adafruit.com/product/4646).

The scary thing is that even these devices are starting to seem big and clunky compared to what's coming down the pike! A few days ago, I was chatting with the chaps and chapesses at Digid (www.digid.com). These young rascals can print nanoscale temperature and force sensors on just about any material, including metals, polymers, ceramics, glasses and semiconductors.

I always think of hypodermic needles as incredibly thin (about 0.5mm in diameter) and highly polished, but they start to look pretty rough and rugged when you zoom in close. The reason I know this is that the folks at Digid showed me a video that made my eyes water (https://youtu.be/UWOAUd1nv4o).

In this video, we zoom in on the tip of a hypodermic needle to discover a sensor that's only 1 micron (1μm or 1/1000th of a mm) long, about 0.1μm (100nm) wide, and so thin it's not worth talking about.

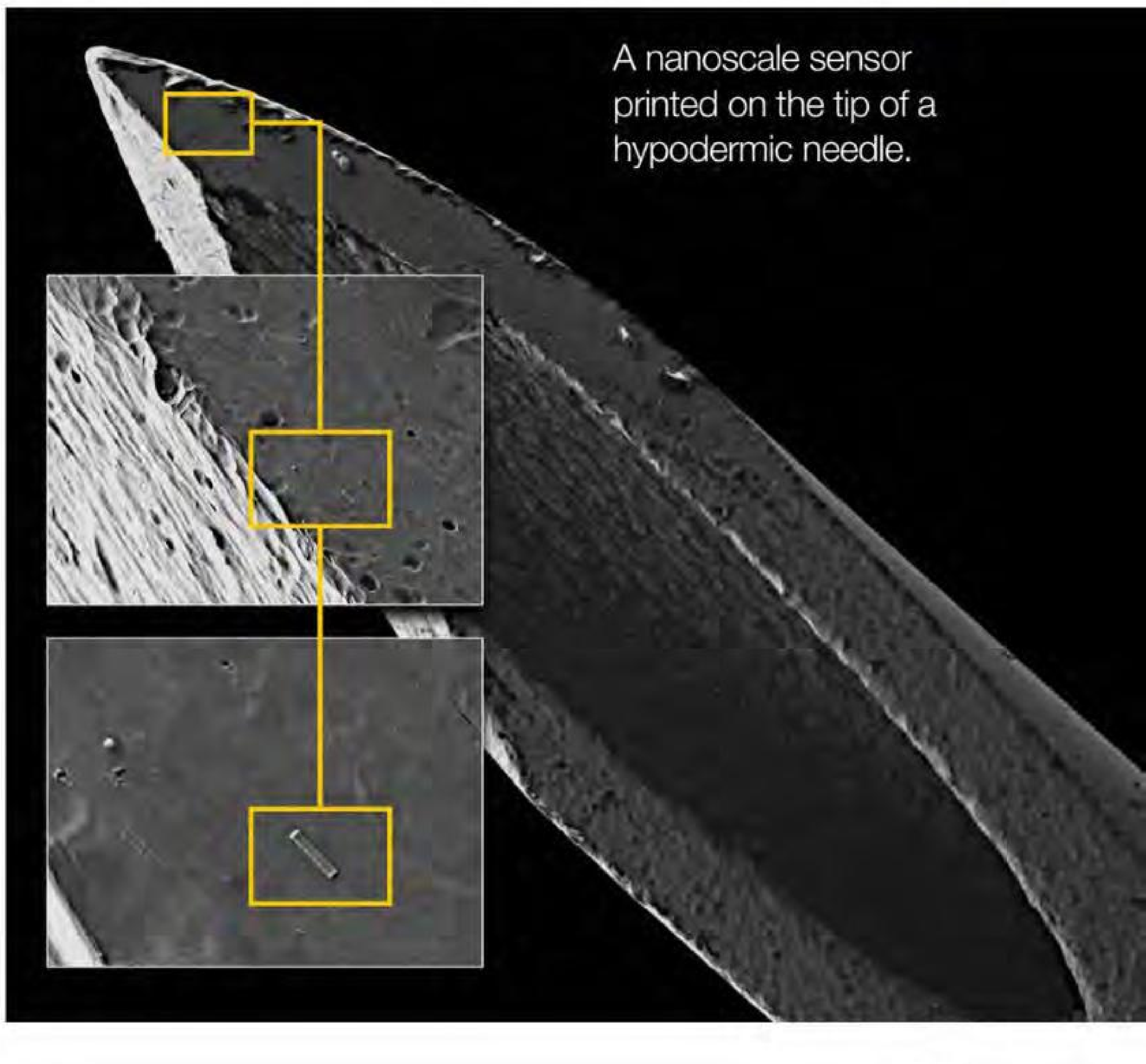
Do we really need sensors this small? Well, the folks at Digid also showed me some examples, such as one of their tiny temperature sensors mounted on the tip of a medical probe (you can be grateful that I'm not going to share the places where this probe is destined to be inserted).

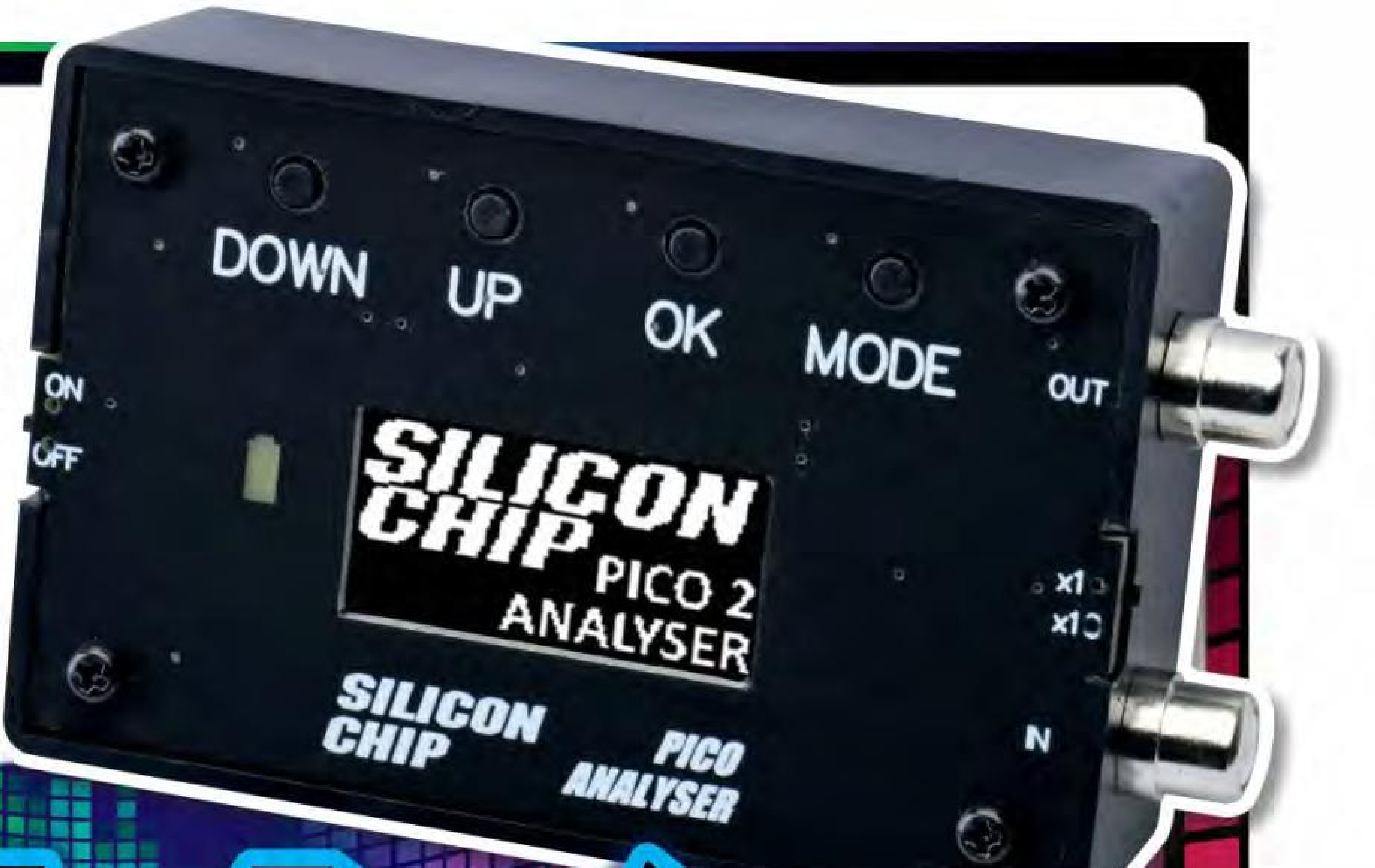
Another example featured one of their force sensors mounted on a scalpel, measuring the forces used to penetrate and cut tissues, a technology that may soon be used to train AI-powered surgical robots.

The wonder years

So here we are, surrounded by devices that are smaller, smarter, quieter and vastly more capable than anything our younger selves could have imagined, yet often sealed, abstracted, and inscrutable.

All this makes me think: we're still only a quarter of the way through the 21st century. Who knows what wonders the coming years will bring? PE


A nanoscale sensor printed on the tip of a hypodermic needle.



Project by Tim Blythman

This handy little tool uses an inexpensive Raspberry Pi Pico 2 microcontroller board and not much else to generate and analyse audio signals. It has oscilloscope and spectrum modes and can run a sweep to plot a frequency response or perform harmonic analysis to check signal quality.

PICO 2 Audio Analyser

The Pico 2 Audio Analyser is a compact handheld device that's powered by an internal rechargeable battery. It can generate and analyse basic audio signals and is suitable for various tasks such as checking amplifiers, wiring, filters etc.

It's a handy tool for working in the field, and for troubleshooting and tinkering with audio circuits. You can even hook it up to a breadboard to test simple circuits like RC filters.

It includes a signal generator, oscilloscope and spectrum displays and can perform harmonic and sweep frequency response analyses.

One potentially interesting use is to monitor the distortion of the mains waveform, which theoretically is a sinewave, but often looks little like one! To do that, you'd connect the output of just about any AC plugpack to its input and put it in distortion analysis mode.

The Pico 2 Audio Analyser uses a Fourier transform to examine the frequency components of a signal. That allows us to create a spectrum display, perform a sweep analysis and calculation a distortion percentage.

Features & Specifications

- Audio signal generator (up to 3V peak-to-peak/1.06V RMS) with selectable frequency
- Sine, square, triangle, sawtooth and white noise waveforms
- Audio signal input with switchable 3.6V and 34V peak-to-peak ranges (1.27/12V RMS)
- Oscilloscope and spectrum displays
- Harmonic analysis with THD measured down to **0.2%** (1.2V RMS, 1.2kHz)
- Can measure and monitor mains distortion with a suitable plugpack
- Sweep analysis with frequency response display
- RCA sockets for input and output
- Runs from USB power or an internal rechargeable battery
- Uses 128×64 OLED display and pushbutton controls
- Compact and portable
- Controllable from a virtual USB serial port
- Typical current draw around 50mA
- Operates for around 12 hours with a fully charged 600mAh battery

Design

When planning this design, we had in mind that it should be inexpensive and compact. The circuitry fits in the smallest Jiffy box (UB5), measuring just 83 × 54mm. The front panel is also the back of the main PCB, recessed into the top of the box, meaning that the height is just 28mm and even less than it would be with the box's included lid.

The display is a 1.3-inch (33mm) diagonal OLED, about the smallest type of display capable of showing graphics. It can also display multiple lines of text. We used this sort of screen in the Advanced SMD Test Tweezers project that was published in the February & March 2024 issues.

No expensive ADC (analog-to-digital converter) or DAC (digital-to-analog converter) chips are used in this design. Instead, a Pico 2 microcontroller board uses its onboard 12-bit ADC to sample the input and a filtered PWM (pulse width modulation) peripheral to drive the output.

The Pico 2 also has an onboard 3.3V switchmode regulator that can operate in PFM (pulse frequency modulation) and PWM modes. We use the PWM mode, as the PFM mode can introduce low-frequency artefacts under the light load levels that this circuit draws.

A clean 3.3V rail is important for

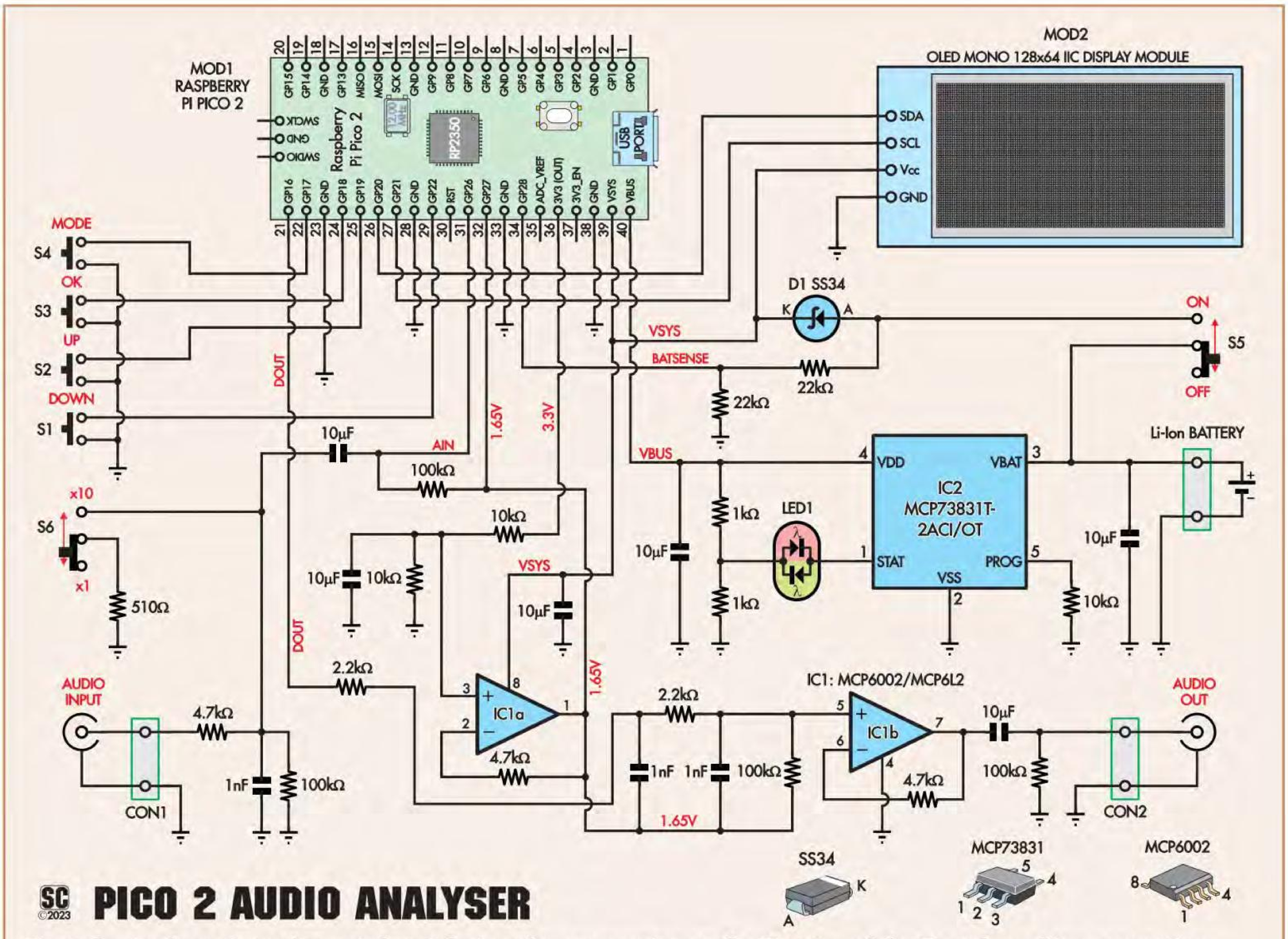

MOD1
RASPBERRY PI PICO 2
Raspberry Pi Pico 2
RP2350
USB PORT
SWCLK
GND
SWDIO
MOD2
OLED MONO 128x64 IIC DISPLAY MODULE
SDA
SCL
Vcc
GND
MODE
S4
OK
S3
UP
S2
DOWN
S1
x10
S6
x1
510Ω
DOUT
AIN
1.65V
3.3V
VSYS
BATSENSE
VBUS
D1 SS34
K
A
22kΩ
22kΩ
ON
S5
OFF
Li-Ion BATTERY
IC2
MCP73831T-2ACI/OT
VDD
VBAT
STAT
PROG
VSS
LED1
1kΩ
1kΩ
10μF
10kΩ
100kΩ
10kΩ
2.2kΩ
4.7kΩ
1nF
IC1a
IC1b
IC1: MCP6002/MCP6L2
AUDIO INPUT
CON1
AUDIO OUT
CON2
1.65V
SS34
MCP73831
MCP6002
PICO 2 AUDIO ANALYSER
SC ©2023

Fig.1: the Analyser is implemented mainly in software running on the Pico 2 module. By only making connections along one side of MOD1, we can mount the Pico 2 on its edge, saving PCB space.

this application, as it is used to set the output level and as the reference for the ADC. While a switchmode regulator is not the best choice for high-quality audio, using the Pico 2's onboard regulator also removes the need to provide separate circuitry and saves us the cost of a separate regulator and its associated components.

Circuit details

Fig.1 shows the circuit diagram of the Pico 2 Analyser. MOD1 is the Raspberry Pi Pico 2 microcontroller module. The input stage of the Pico Analyser receives a signal via the RCA socket at CON1. A 4.7kΩ resistor combined with a 1nF capacitor gives a low-pass filter with a -3dB point of around 34kHz. This reduces any high-frequency components that the ADC might alias.

The 100kΩ resistor keeps this signal biased to ground whenever nothing is connected. A 10μF capacitor AC-couples the signal so that the Pico 2's analog input pin (AIN, pin 31) can be DC-biased to 1.65V (half of the 3.3V supply) by another 100kΩ resistor.

The incoming network attenuates the audio slightly, to around 91% of its original level. That means that voltages up to 3.6V (peak-to-peak) can be measured before clipping occurs, corresponding to around 1.2V RMS.

Switch S6 can connect a 510Ω resistor in parallel to the first 100kΩ resistor, changing the divider formed with the 4.7kΩ resistor. This allows levels up to 34V peak-to-peak (or 12V RMS) to be measured without clipping. This resistor does change the filter characteristics and may let in more higher-frequency components than the ×1 range.

The output signal from the Pico 2 (DOUT, pin 21) is a PWM signal at around 250kHz, so it first passes through two RC filter stages, each consisting of a 2.2kΩ resistor and 1nF capacitor. A 100kΩ resistor also biases this to the 1.65V rail so that a known level is present if the pin is not being driven.

The two RC stages give a similar -3dB point to the input stage, attenuating the 250kHz PWM artefacts by 24dB in total, compared to the 12dB for a single stage. The result is not hifi, but good enough for our purposes.

This signal is buffered by IC1b, then AC coupled and biased to ground by another 10μF/100kΩ pair before being made available at CON2.

The filtering and biasing mean that around 3.1V peak-to-peak is available from a 3.3V rail, or about 1.1V RMS, although this is limited by the op amp's drive near its rails and will depend on the output load.

Buttons S1-S4 are connected to other available digital input pins. These are used to provide controls for the user interface. Internal pullup currents supplied by the Pico 2 hold the corresponding pins high unless the switches are closed, pulling the attached pins to ground.

Power supply

Important to the circuit's operation is a schottky diode internal to MOD1, from its VBUS pin (40) to its VSYS pin (39).

The Pico 2's switchmode regulator is fed from VSYS and its output is available at the 3V3 pin. You'll note that only one side of the Pico 2 has connections. By mounting it on its edge against the PCB, we save much PCB space and it fits more easily in the box.

The circuit can be powered from a USB supply via the Pico 2's onboard USB connector, leaving around 4.7V available at the VSYS pin. Alternatively, power from a rechargeable lithium battery is provided via D1 when switch S5 is closed, giving around 3.4-3.9V at VSYS. The regulator on the Pico 2 can handle between 1.8V and 5.5V, so these are all comfortably within its operating range.

When USB power is available, the battery is charged by IC2. The two 10µF capacitors provide the input and output bypassing it requires, while a 10kΩ resistor between its pin 5 (PROG) and ground sets the charge current to 100mA.

The STAT pin (pin 1) is low during charging and goes high when charging is complete, so the bi-colour LED will show the charging state: red during charging or green when charged; separate 1kΩ resistors limit the LED currents.

MOD2 is a 1.3-inch (33mm) OLED display module. Its VCC pin is fed with whatever voltage is available at the Pico 2's VSYS pin, and its onboard regulator provides 3.3V for its operation, as well as internal pullups for the I²C control lines, SDA and SCL.

The I²C lines are taken back to the appropriate pins on the Pico 2 so it can update the display.

IC1 is a low-voltage dual op amp, and it too is fed from the VSYS rail with a 10µF supply bypass capacitor. Since VSYS is slightly higher than 3.3V, this provides a bit more headroom than the 3V3 rail would allow.

The 3V3 rail is divided by a pair of 10kΩ resistors and bypassed by the 10µF capacitor to give the 1.65V mid-rail reference. This is buffered by unity-gain op amp IC1a.

The Pico 2 has four ADC channels, with one internally connected to VSYS via a divider, so two are left after we've fed in our audio signal. We've connected one of these to the 1.65V rail so the Pico 2 can check that it is correct.

DOWN
UP
OK
MODE
OUT
IN
SILICON CHIP
PICO ANALYSER

The right-hand end of the case has two holes for the RCA sockets and a notch for S6.

The remaining ADC channel is connected to a divider comprising two 22kΩ resistors across the battery downstream of the switch. This allows it to read the battery voltage when S5 is closed, ensuring the battery is not drained when the unit is switched off.

Software

We used the Arduino IDE to create the software, mainly because so many libraries are available. We use OLED libraries from Adafruit that make generating the graphics needed for the spectrum, oscilloscope and frequency response modes easy.

The audio generation software is a fairly straightforward PWM implementation, where the PWM duty cycle is updated between samples to provide a varying waveform. It is based on the software we wrote for the Pico BackPack, which has a stereo audio output (published in the March 2023 issue).

While we're using 8-bit PWM, the data is calculated and stored as 16-bit samples, with the PWM data derived from the upper eight bits. Then the remainder due to the lower eight bits is dithered over several PWM cycles per sample period, slightly improving the effective resolution.

A block of samples equivalent to about 200ms is generated to provide this data. For all but the lowest frequencies, this means that the frequency does not need to divide evenly into the sampling rate since the sample block contains multiple cycles.

We use the second processor core to calculate and update the dithered samples. That is about all the second processor does, so not much can interrupt audio generation once it is running.

A similar technique is applied to the analog input sampling. The 12-bit ADC runs at 490kHz, very close to its maximum speed of 500kHz. The DMA peripheral captures a block of samples over about ⅒th of a second without interrupting either processor. This means we can detect frequencies down to around 10Hz.

The Pico 2's ADC is improved over that of the Pico; the RP2040 chip used in the Pico suffers from an issue in its ADC module that affects its accuracy and it was only capable of providing nine effective bits of resolution. The Pico 2's revised ADC no longer has this problem this and so provides better resolution.

The oscilloscope mode uses the raw samples for its display, which provides adequate resolution for the 50-pixel vertical axis. The other modes apply downsampling before running a Fourier transform to extract the frequency elements of the sampled waveform. Much of the software is involved in drawing the various displays and user interfaces.

Construction

First, use the blank PCB to mark the box, then perform the cuts shown in Fig.2. One end of the box has a notch for

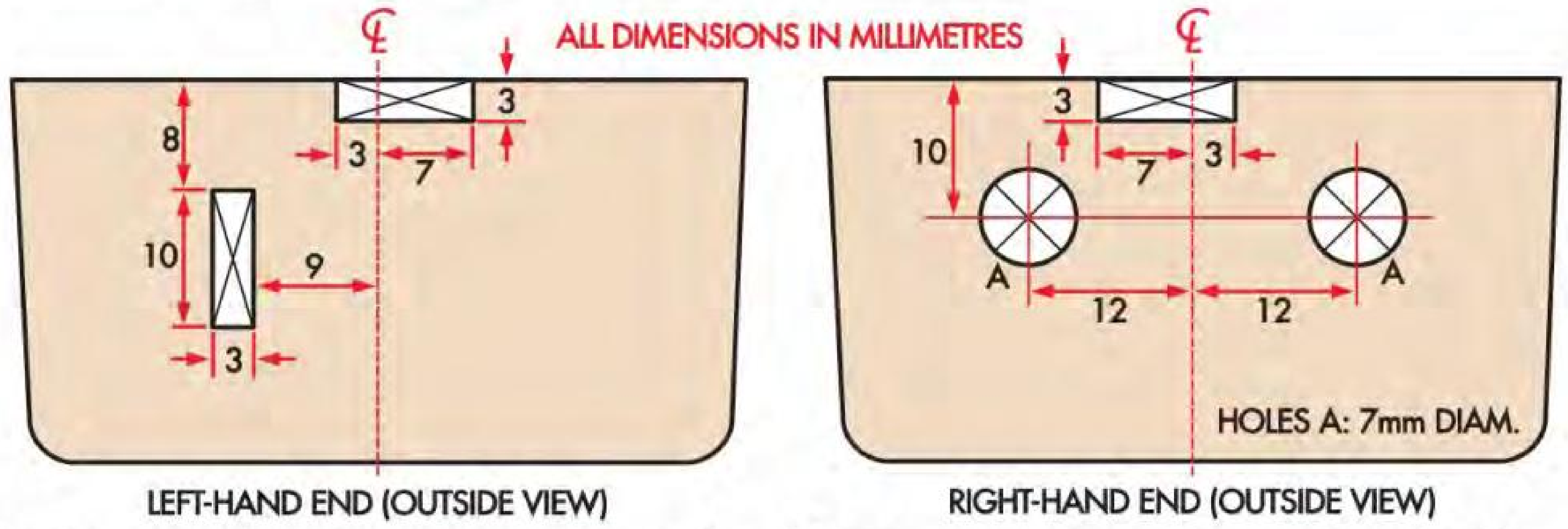
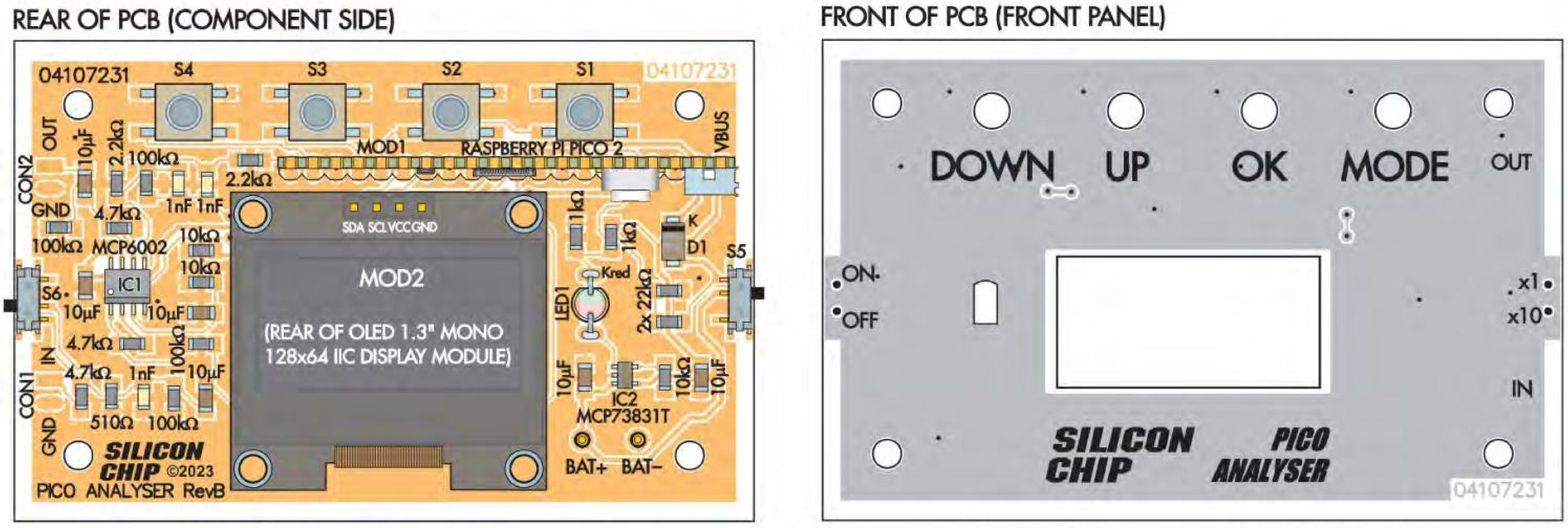

ALL DIMENSIONS IN MILLIMETRES
℄
℄
8
3
3
7
10
9
3
LEFT-HAND END (OUTSIDE VIEW)
10
3
7
3
A
A
12
12
HOLES A: 7mm DIAM.
RIGHT-HAND END (OUTSIDE VIEW)

Fig.2: many of these cuts can be made without measuring. The notches for the switches at the top of the box can be marked using the PCB as a template, while the holes for the RCA sockets do not need to be precisely located, as the sockets are wired with flying leads.


REAR OF PCB (COMPONENT SIDE)

FRONT OF PCB (FRONT PANEL)

Fig.3: the rear of the PCB also forms the device's front panel, so all components are surface-mounting. It's a bit cramped with the OLED and Pico adjacent. We recommend fitting the Pico first and ensuring it is aligned with the hole for its USB socket, then fit the OLED and check its operation before continuing. Note that the OLED is mounted face-down on the rear side.

S5 and the Pico 2's USB socket, while the other has a notch for S6, plus two holes for the RCA sockets.

The notches can be marked using the edge of the PCB, which might be easier than using a ruler to find the midpoint.

Make a pair of vertical cuts on each side, not quite to the desired depth, with a sharp hobby knife or hacksaw. Score along the bottom of the notches with a sharp knife, then carefully flex the tab, which should break off along the scored line. Tidy the corners and edges to the correct depth with a small file or sharp knife if necessary.

Mark out the slot for the USB socket and start by drilling two or three holes inside the lines. Then use a small file or sharp-pointed hobby knife to square up the edges of the slot. We can use this slot later to align the Pico 2 correctly, or alternatively, we can make the Pico 2 fit it more easily than we can adjust the hole!

The RCA sockets mount in drilled holes that can be made with a twist or step drill. Their exact positions are not critical, as the sockets are connected by flying leads. The measurements shown match our prototype and work well.

Starting with a smaller 3mm pilot hole will make it easier to align the holes and adjust them if they are not aligned. We've specified 7mm holes to suit the RCA sockets we've used, but check if a different size is required for your parts.

PCB assembly

Many of the components are fairly large standard SMDs. There are a few parts that are mounted in a slightly unorthodox fashion. We recommend starting by fitting the SMDs; you will need a fine-tipped iron and solder, flux paste, tweezers and good illumination. Some solder-wicking braid will be handy, as will some solvent to clean up any excess flux.

Use fume removal (such as a fume removal hood) to ensure you are not exposed to smoke from the flux. If that is not possible, work outside in fresh air.

Refer to the Fig.3 overlay diagram for the component placements and orientations. You should also consult the photo showing the PCB fitted with surface-mounting parts.

The components are pretty close together, and IC2 is the smallest part, so start with it. Put some flux paste on the pads and align the five pins with them; they will only fit one way. Tack one lead on the side with two pins and check that all the other pins are within their pads, adjusting as necessary. Solder the remaining pins and then go back to refresh the first pin.

Check for bridges and use solder wick and fresh flux to draw excess solder away, if necessary.

Use a similar technique for IC1. Its pins are more widely spaced, so soldering should be easier. Make sure pin 1 of IC1 (which might be marked with a bevel along one edge) is aligned to the dot on the PCB silkscreen.

The capacitors are spread around the PCB. Be sure not to mix up the two values, although the 10µF parts will probably be thicker than the 1nF parts.

The different resistor values all need to go in the correct locations too. For these passives, use the same basic soldering technique. Solder one lead, then check and adjust before soldering the other lead.

The single diode is a bit larger, and you must ensure its polarity is correct, with its cathode stripe towards the "K" on the PCB. If this is reversed, you risk connecting the battery directly to the USB supply, which will probably cause something to burn out.

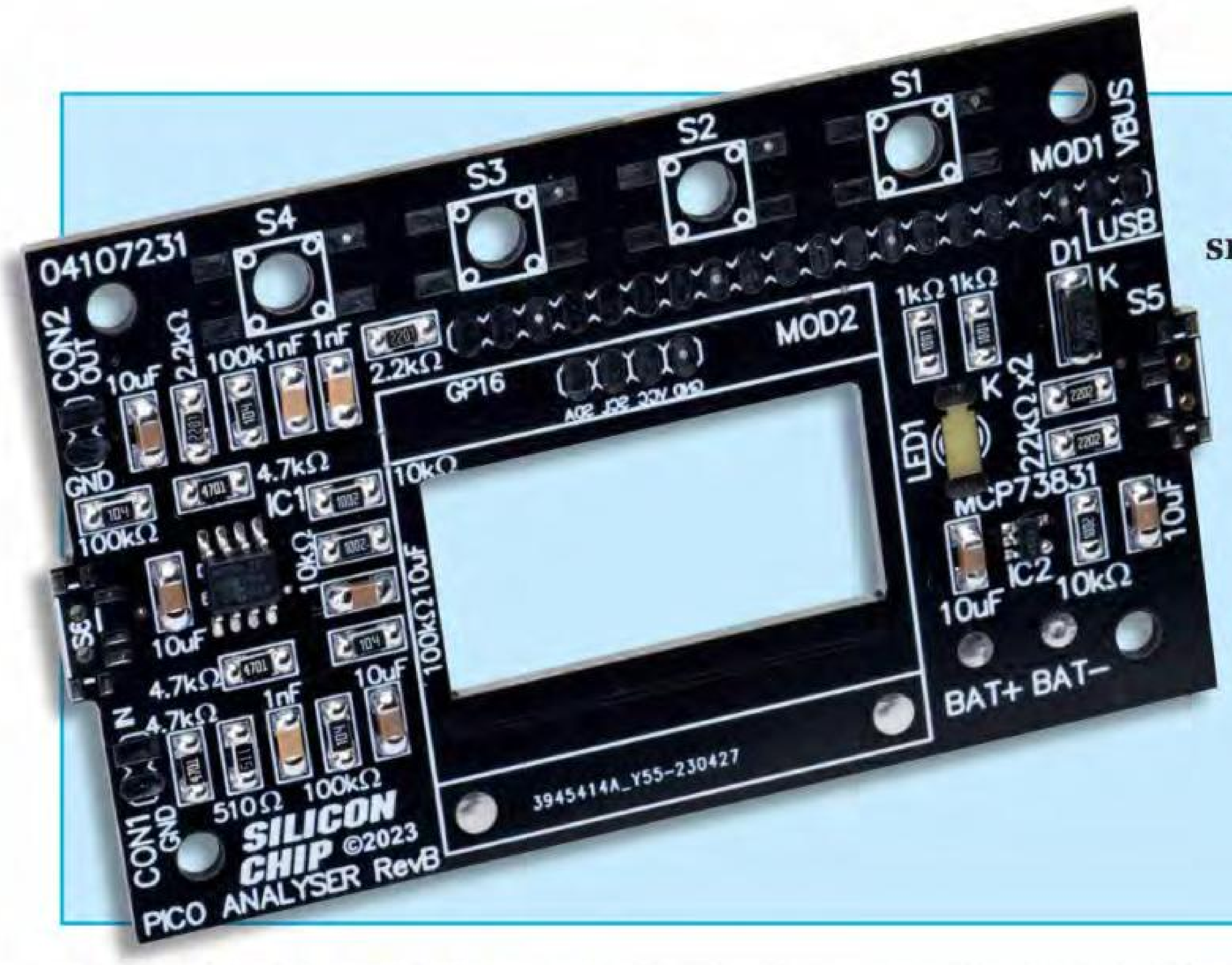
Now is a good time to clean the PCB with a flux solvent. Doing so now avoids the possibility of solvents getting into the switch mechanisms. Isopropyl alcohol is a good all-round choice.

Allow the PCB to dry thoroughly before continuing.

Fit slide switches S5 and S6 next. They have small leads but are easy to align as they have locating pins in their

DOWN
UP
OK
MODE
OUT
ON
OFF
x1
x10
IN
SILICON CHIP
PICO ANALYSER

The front-facing side of the PCB.



Use this photo as a guide to fitting the smaller components. This stage of assembly is a good point to clean off any excess flux in preparation for adding the final components like the switches, LED, Pico and OLED.

undersides that lock into holes in the PCB. Tack one lead, confirm that they are flat and then solder the other leads.

Next, fit the four reverse tactile switches, S1-S4. We found it helpful to splay the leads out from the bodies so that the switch stems protrude further through the PCB holes. This makes them easier to operate.

After soldering one pin, it's also a good idea to check that the switch stems are centred in their holes through the PCB. That will ensure the front panel looks good and eliminate the possibility of the stems jamming on the PCB.

Once you're happy with them, solder the other three pins on each switch. Be sure to use a generous amount of solder to ensure that they have good mechanical strength.

The tricky bits

The LED is mounted unusually. While bi-colour SMD LEDs are available, they often have independent leads for the two LEDs, making the pads small and tricky to solder, so we're using a 3mm through-hole LED as a reversed surface-mounting device.

The pad marked K corresponds to the cathode of the red LED inside such a device. If you're unsure and don't have the means to test it, just fit the LED one way; if it is incorrect, swap the leads.

Carefully bend the leads by 180° and trim them so they are slightly longer than the LED lens. As you can see from our photo overleaf, the tip of the LED is pointed at the opening in the solder mask (facing towards the PCB). Solder the LED leads to the two pads.

Fine-tipped tweezers will help to position the component until one lead is soldered. Solder the other lead, then refresh the first joint.

The next part is the Pico 2 module. Before proceeding, check that the PCB (with S5 and S6 mounted) sits flush and slots neatly into the box's notches. The top of the PCB should sit level with the surrounding box. This is to ensure that the USB connector on the Pico 2 can align correctly with the slot in the box. Adjust the notches in the box if necessary.

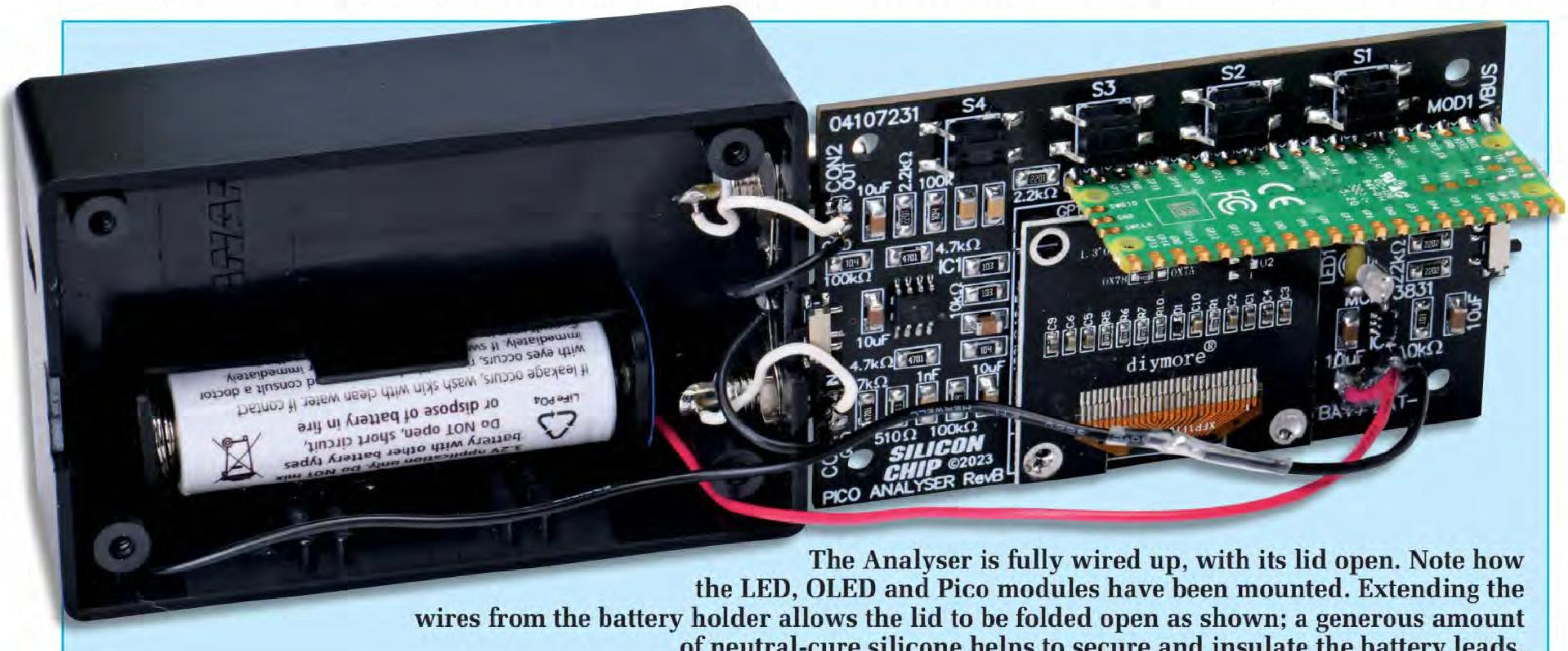
Working with just one end pad on the Pico 2, tack it roughly into place at right angles to the main PCB. The Pico's PCB should sit back slightly from the edge of the main PCB, with the USB connector protruding slightly.

Note its relative orientation, with the VBUS pin closest to the edge of the PCB and GP16 at the other end. The USB socket should be above the corresponding marking on the silkscreen too.

Tack one pad at the other end and carefully adjust the Pico 2 to be at right angles to the main PCB. Test it in the box and see that it is aligned with the slot. Remember that the top of the PCB will sit flush with the top of the box.

Once you are happy with the location of the Pico 2, solder the remaining pins. We found it easiest to feed in the solder from the bottom of the Pico 2 (on the side facing the switches) and apply the iron to the other side, ie, the Pico's top. Ensure there is a generous fillet on each of the 20 pins to hold the module securely.

Now cut the LED lead offcuts (or other fine wire) into four pieces, each about 1cm long. It will help if they are all slightly different lengths to stagger their insertion into MOD2's pads. Using the tweezers to hold each one, solder them to the centre of the pads for MOD2. They should sit vertically.



The Analyser is fully wired up, with its lid open. Note how the LED, OLED and Pico modules have been mounted. Extending the wires from the battery holder allows the lid to be folded open as shown; a generous amount of neutral-cure silicone helps to secure and insulate the battery leads.

Remove the protective film from the front of the OLED and place it face-down over these wires and flat against the main PCB. You will see that the backwards markings on the PCB now correspond to the OLED pins.

Finally, solder each wire to the OLED. Be gentle, as there is little more than surface tension holding the pin in place to the main PCB. You might need to adjust the wire with tweezers.

Verify that the OLED is accurately aligned with the silkscreen markings. If it is not, the misalignment will be evident in use.

Now is a good time to run some quick tests to ensure that the OLED and Pico 2 are correctly soldered, but the Pico will need to be programmed if it is not already.

Programming the Pico 2

Hold the white BOOTSEL button while connecting the Pico 2 to a computer via a USB cable. You might not need to hold the button if you have a new, unprogrammed Pico. Picos supplied in kits are generally not programmed as it's easy for constructors to do.

Then copy the 0410723B.UF2 file to the RP2350 drive that should appear on your computer's file system. If everything is working, you should see the OLED screen light up after a second. If not, go back and check the solder joints and component placement.

Verify that the display contents are square within the PCB cutout. If they are not, you might be able to gently twist the OLED by a small amount.

Completion

With the OLED aligned, use the remnants of the lead offcuts to secure its two lower holes to the matching pads on the PCB. The connection should work much the same as for the four smaller pads on the top of the OLED module.

Prepare the RCA sockets by disassembling them. Cut two pieces of white wire about 4cm long and two pieces of black wire about 4cm long. The colours are not critical, but using two contrasting colours will help identify them.

Solder one end of each of the white pieces of wire to the centre connection of an RCA socket. Similarly, solder one end of each black wire to the washer, which becomes the ground connection.

Parts List – Pico 2 Audio Analyser

1 double-sided PCB coded 04107231, 83 × 50mm, with black solder mask
1 UB5 Jiffy box (83 × 53 × 30mm)
2 chassis-mount RCA sockets (CON1, CON2) **[Farnell 4161690]**
1 single AA cell holder with flying leads
1 14500 (AA-sized) Li-ion rechargeable cell with nipple
1 Raspberry Pi Pico 2 board, programmed with 0410723B.UF2 (MOD1)
1 1.3-inch (33mm) OLED module (MOD2) **[SILICON CHIP SC5026]**
4 reverse-mount SMD tactile switches (S1-S4) **[Adafruit 5410]**
2 SPDT SMD slide switches (S5-S6)
4 M3 washers, 1.5mm thick
2 20cm lengths of hookup wire (eg, white and black)
1 4cm length of fine bare wire (eg, lead offcuts from LED1)
1 small tube of neutral-cure silicone sealant
1 short RCA-RCA cable (for testing & calibration)

Semiconductors
1 MCP6002 or MCP6L2 rail-to-rail dual op amp, SOIC-8 (IC1)
1 MCP73831-2ACI/OT Li-ion charge regulator, SOT-23-5 (IC2)
1 bi-colour red/green 3mm LED (LED1)
1 SS34 40V 3A schottky diode, DO-214 (D1)

Capacitors *(all M3216/1206 size, X7R ceramic)*
6 10µF 16V+
3 1nF 50V

Resistors *(all M3216/1206 size, 1% 1/8W)*
4 100kΩ
2 22kΩ
3 10kΩ
3 4.7kΩ
2 2.2kΩ
2 1kΩ
1 510Ω

Pico 2 Audio Analyser Kit

SC6772 (~£31): includes the PCB and everything that mounts directly on it. The Pico is supplied blank and will need to be programmed using a computer and USB cable.

Assemble the sockets into the holes in the enclosure by securing the washer with the nut. Adjust them such that the wires poke out the top of the box, then bend them over the end of the box.

Next, place the PCB upside down next to the enclosure and solder the wires, as shown in the photos opposite and overleaf. The two black wires go to the GND pads on CON1 and CON2, while the white wires go to the corresponding pads marked IN and OUT. Use a generous amount of solder to ensure a firm connection.

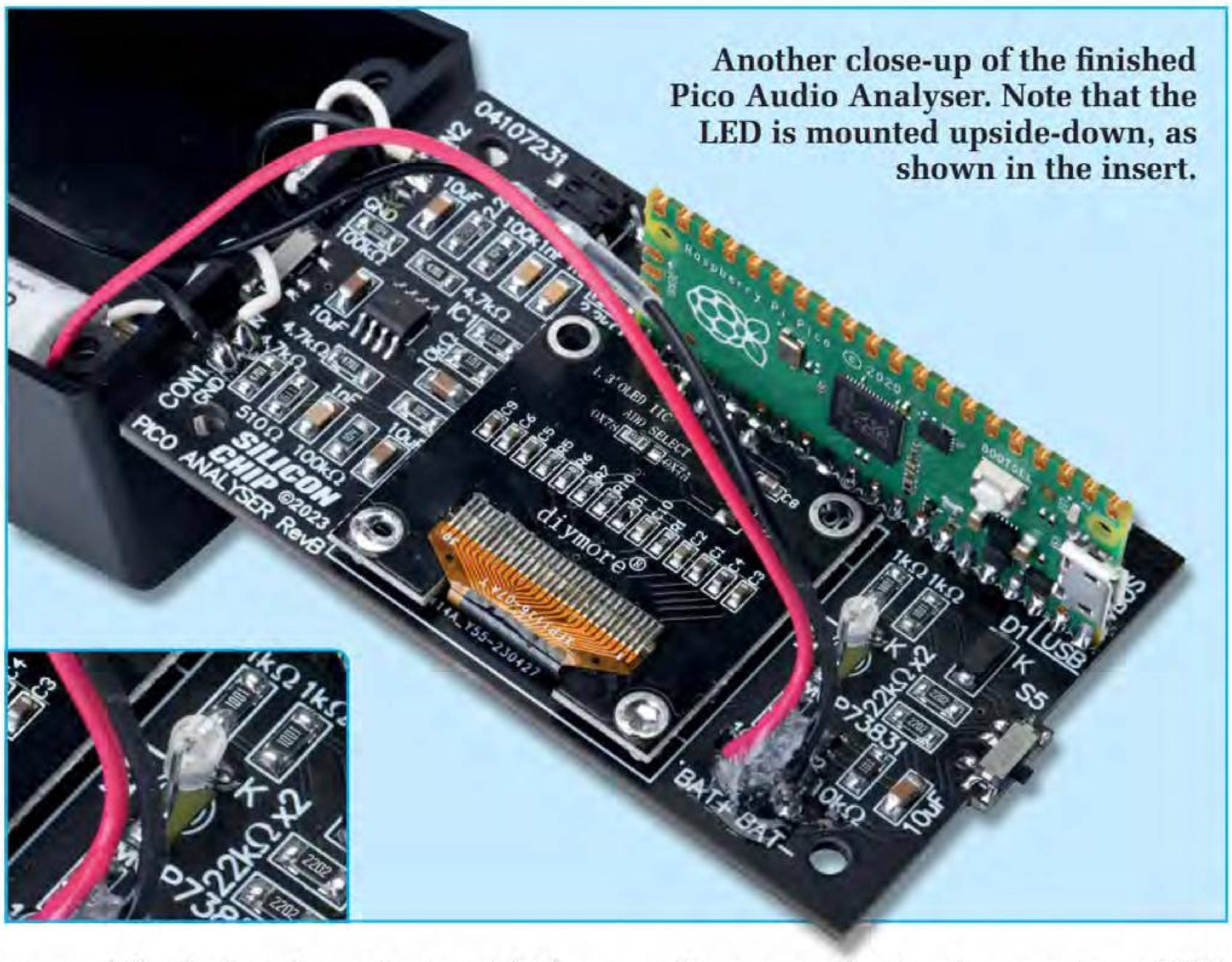
Using neutral-cure silicone or similar gap-filling glue, secure the battery holder to the bottom back corner of the box, with the opening facing outwards. Solder the wires to the BAT+ and BAT- pads on the PCB, being sure to connect the red wire to BAT+ and the black wire to BAT-.

On our prototype, we slightly extended one of the battery leads to allow the PCB to fold fully open away from the box. That simplified testing and assembly.

Like the RCA sockets, use a generous solder fillet to secure the battery leads. Apply silicone around the BAT+ and BAT- pads to further secure the battery leads and insulate any bare wire. We do not want these to come loose, as there is a good chance that their bare ends would cause the battery to be short-circuited.

While waiting for the silicone to cure, you might like to also add some more to CON1 and CON2 on the PCB to secure the audio connections, as well as any exposed metal on the outside of the battery holder itself (for example,

If you do not have an RCA-RCA cable, a simple loopback cable like this can be made by soldering a short wire between the centre pins of two RCA plugs. Such a cable is necessary for testing and calibrating the Analyser. We also found it handy to have a pair of RCA plugs fitted with jumper wires to allow connecting to a breadboard for experimentation.



Another close-up of the finished Pico Audio Analyser. Note that the LED is mounted upside-down, as shown in the insert.

around the battery's contacts with the holder).

After the silicone has cured fully, fit the battery, making sure to check its polarity. Switch the unit on with S5 and confirm that the OLED illuminates after about a second. If it does not, remove the battery and check for any problems.

Apply power to the USB socket and see that the LED lights up red initially and then goes green when the battery finishes charging. If the LED starts green, it might be reversed. It's a good idea to remove the battery before making any changes to the circuit.

Four washers sit between the PCB and the box's pillars to keep the PCB flush with the top of the box, so thread these over the screws as you screw them into the box. Take care that you don't pinch any of the wires.

Calibration

After some calibration steps, the Analyser will be ready to use. The Analyser will function without calibration, but its accuracy will not be as good.

You'll need a multimeter or oscilloscope that can accurately measure a 500mV AC RMS signal and an RCA plug to RCA plug cable to make a loopback connection between the input and output.

We used a pair of RCA plugs with a short piece of wire connecting their centre terminals (the ground connection is made via the PCB in this case).

Power up the Analyser using a USB cable to give the battery a chance to charge. The splash screen shows for a few seconds as the bias voltages stabilise.

Press the MODE button until the SETTINGS screen appears (Screen 1), then press OK. Screen 2 shows the first calibration item, the INPUT OFFSET. Ensure nothing is connected to the input and wait until the value seen on the fourth line settles to a steady value and press the DOWN button, then OK.

Screen 3 is the OUTPUT LEVEL calibration. The Analyser will deliver a nominal 500mV RMS sinewave, which should be measured at the CON2 output. Use the UP and DOWN buttons to adjust the calibration ratio until your meter reads 500mV, then press OK.

Screen 4 sets the INPUT LEVEL for the ×1 range. Connect the CON2 output to the CON1 input and set switch S6 to the ×1 position. Since the Analyser knows it should be receiving a 500mV signal, it can calculate the calibration ratio easily.

When you see the "DOWN to set" message, press the DOWN button to load the calculated ratio. This allows us to check that a valid signal is used for the calculations.

If you don't see this message and are sure that S6 is set correctly, there could be a minor problem with the PCB, such as a resistor being the wrong value. This step also depends on the 500mV reference being set correctly in the previous step. Press OK to proceed to the next screen.

Screen 5 sets the INPUT LEVEL on the ×10 range, so leave the cable connected and change S6 to the ×10 range. You will see a prompt similar to the previous step; press DOWN when it appears.

Screen 6 is used to save those parameters to flash memory; press DOWN to do so, and you should see a message reporting that this has occurred. If the settings have somehow become corrupted, you can use the UP button here to restore the defaults.

Operation

The remaining screens show the operating modes. The MODE button cycles between the modes, while the UP, DOWN and OK buttons provide controls within each mode. Generally, a pair of angle brackets <> highlights the value being changed.

When switching between ×1 and ×10, the input mode must also be manually changed on the SETTINGS page to match.

Pressing DOWN selects the ×1 mode (and uses the ×1 calibration factor), while pressing UP selects the ×10 mode. The last line of this page shows the current scaling.

The top right corner of the SETTINGS page shows the battery voltage when the power switch (S5) is turned on. Take care that the Analyser is not left switched on when not in use, as there is nothing to prevent the battery from being overdischarged.

The first mode (seen in Screen 7) controls the WAVE OUTPUT. This will continue to run at its last setting unless another mode needs to take control of the output. This can occur when a SWEEP is run, or the SETTINGS needs to produce its calibration waveform.

The OK button cycles between the various parameters, while the UP and DOWN buttons change them.

The set frequency can vary between 10Hz and 10kHz; the frequency steps are smaller for lower frequencies. Since the Pico 2 has a crystal oscillator, we have provided no frequency calibration adjustment. The frequency accuracy of the crystal is around 30ppm (0.003%), which is good enough.

The output level can be set in steps of 50mV as either peak-to-peak or RMS, and the corresponding equivalent values are displayed depending on what is selected. The ratio between the peak-to-peak and RMS values changes depending on the waveform.

Values up to about 2V peak-to-peak should give clean outputs before op amp drive limits come into play, depending on the output load. The chosen op amp is quite robust and can handle an output short circuit indefinitely.

The next option cycles between sine, square, triangle, sawtooth and white noise waveforms, while the last option allows the signal to be turned off or on without changing any other settings. Internally, the Analyser generates a 0V amplitude waveform when the output is off.

The SPECTRUM mode displays the spectrum of the input waveform (Screen 8). The UP and DOWN buttons change the horizontal scale, while the OK button switches the vertical scale between the PEAK and TOTAL (RMS) amplitude.

As the Fourier Transform includes a windowing step, even a pure sinewave will typically be spread across multiple frequency bins. The Low Frequency Distortion Analyser article from the April 2016 issue has more information about windowing.

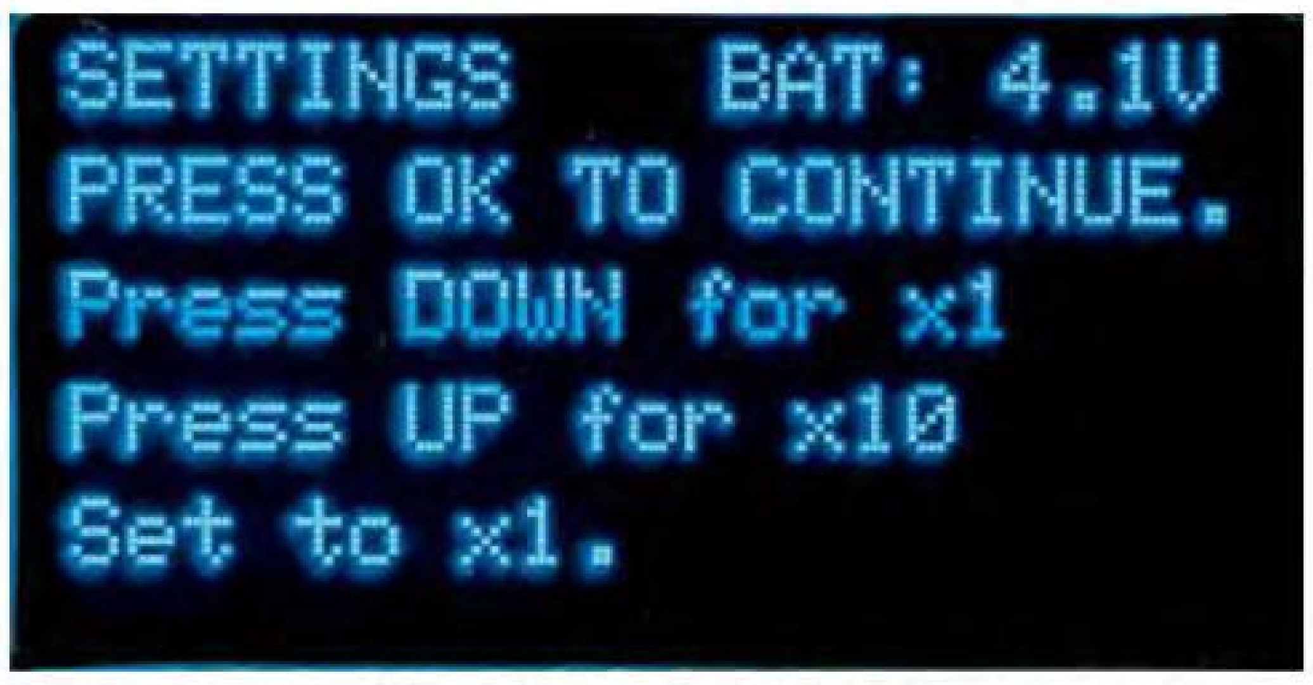
The calculated peak frequency is interpolated between the bins and may also be slightly off due to rounding errors.

The SCOPE mode (Screen 9) simply shows the shape of the waveform as you would see on an oscilloscope. The UP and DOWN buttons change the horizontal (time) scaling, while the OK button toggles between dots and lines for the plot. You might find the line mode clearer when many cycles are displayed.

The vertical scaling is automatic and based on the amplitude, shown as a peak-to-peak value on the left. The SCOPE attempts to trigger on a positive-going zero crossing and, if not, will simply display the last part of the sample it has taken.

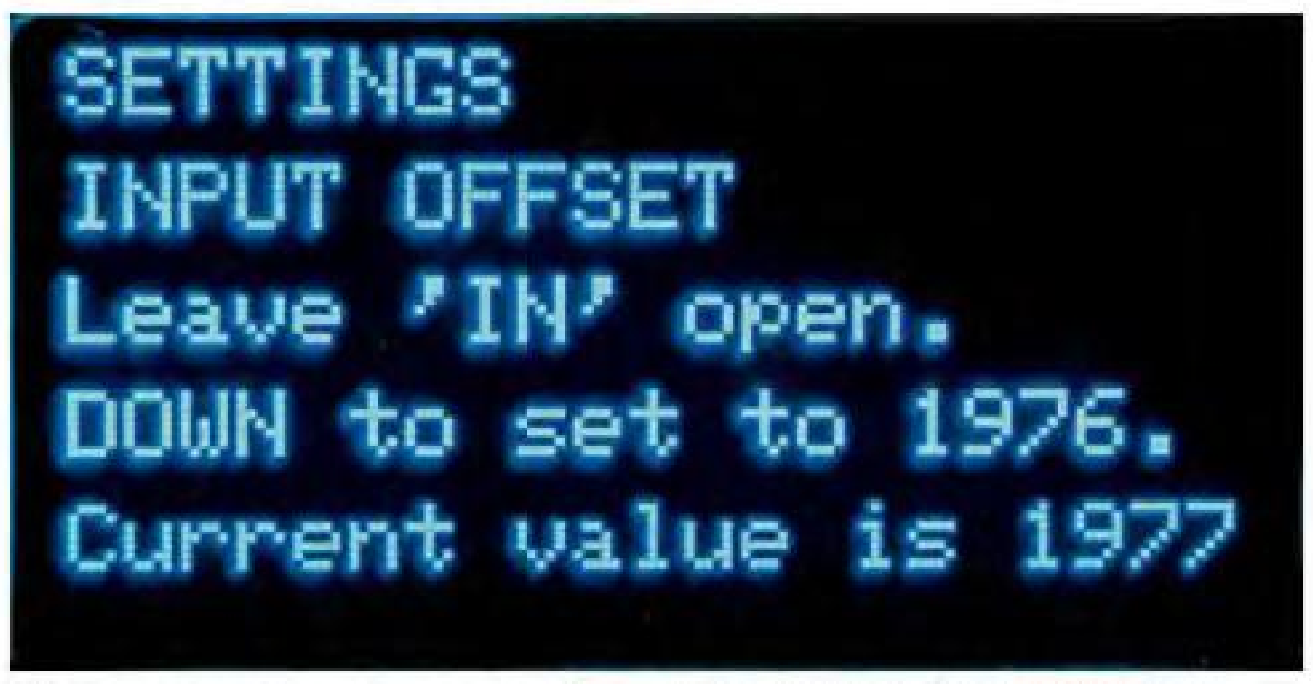
Harmonic analysis

HARMONIC ANALYSIS (Screen 10) provides information about the detected fundamental frequency, an analysis of the harmonics and the measured THD (total harmonic distortion). The UP, DOWN and OK buttons do nothing in this mode.



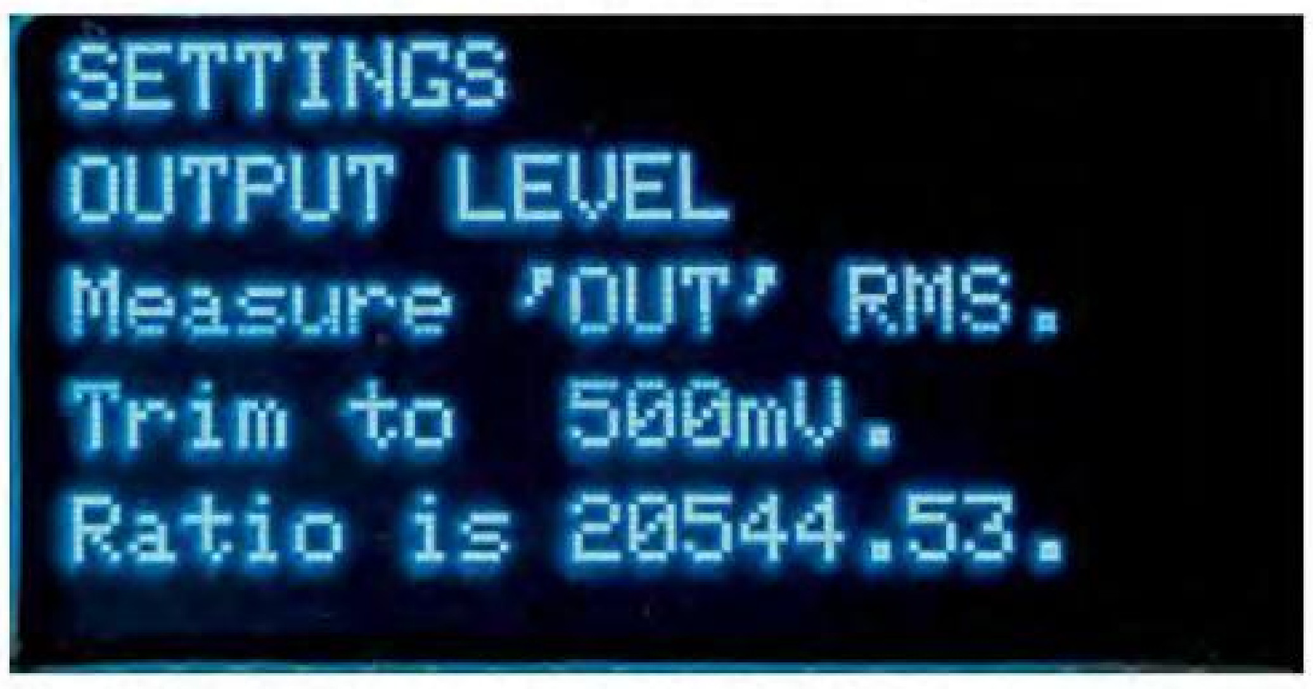
```
SETTINGS    BAT: 4.1V
PRESS OK TO CONTINUE.
Press DOWN for x1
Press UP for x10
Set to x1.
```

Screen 1: when the Analyser is first powered up, use the MODE button to cycle through to the SETTINGS page to perform the calibrations. Press OK to start the process.



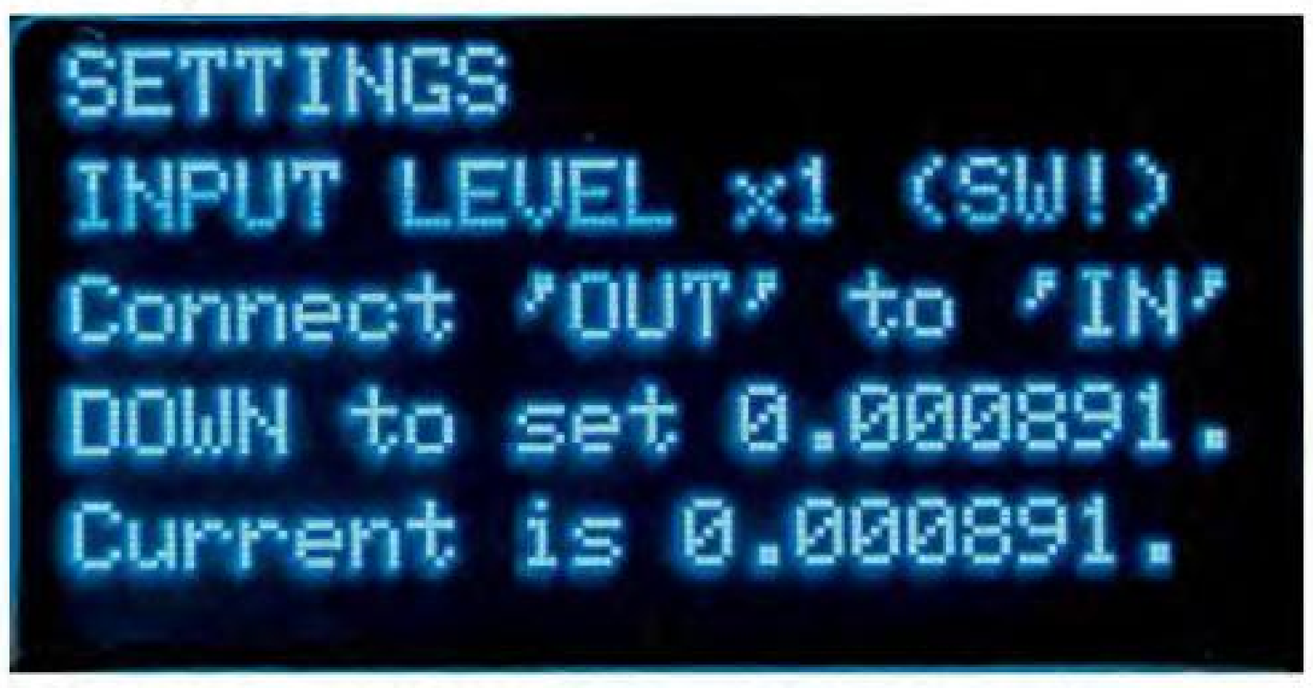
```
SETTINGS
INPUT OFFSET
Leave 'IN' open.
DOWN to set to 1976.
Current value is 1977
```

Screen 2: to set the INPUT OFFSET, leave the input open and allow the displayed level to settle to a steady value. Then press the DOWN button to store this value, followed by OK.



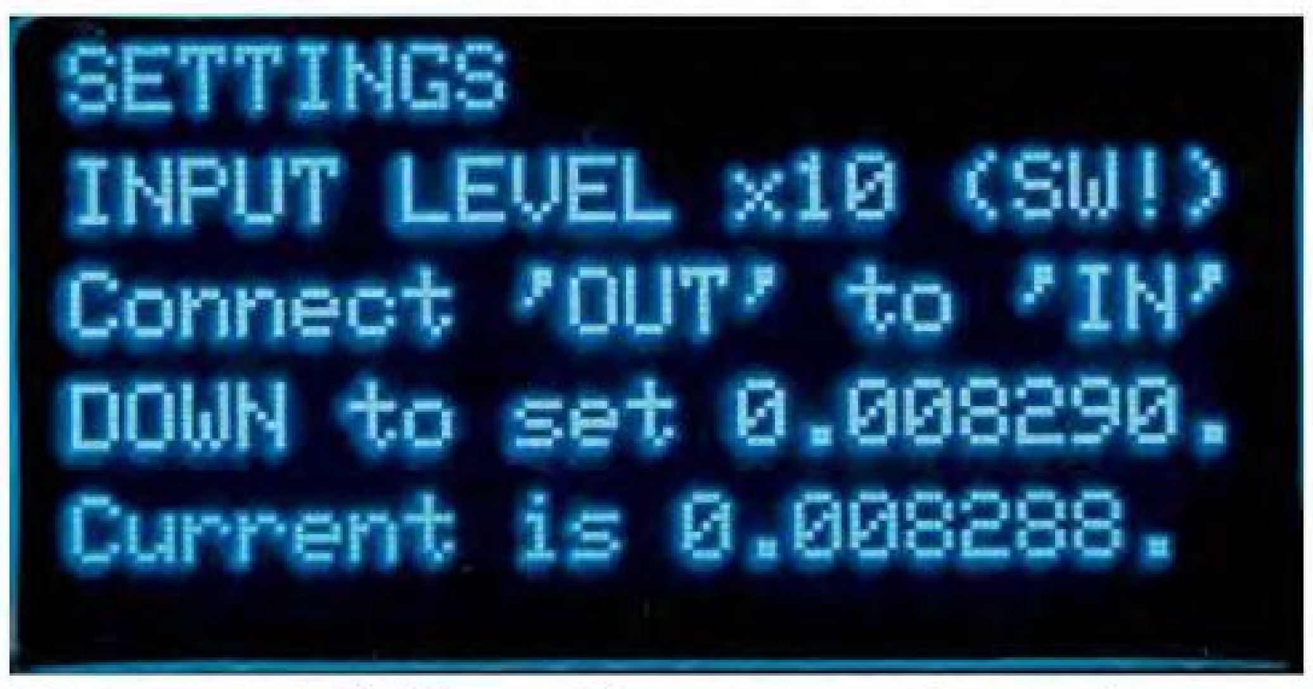
```
SETTINGS
OUTPUT LEVEL
Measure 'OUT' RMS.
Trim to 500mV.
Ratio is 20544.53.
```

Screen 3: use an AC RMS meter or similar instrument to measure the output and adjust (using UP and DOWN) until the meter reads 500mV, then press OK.



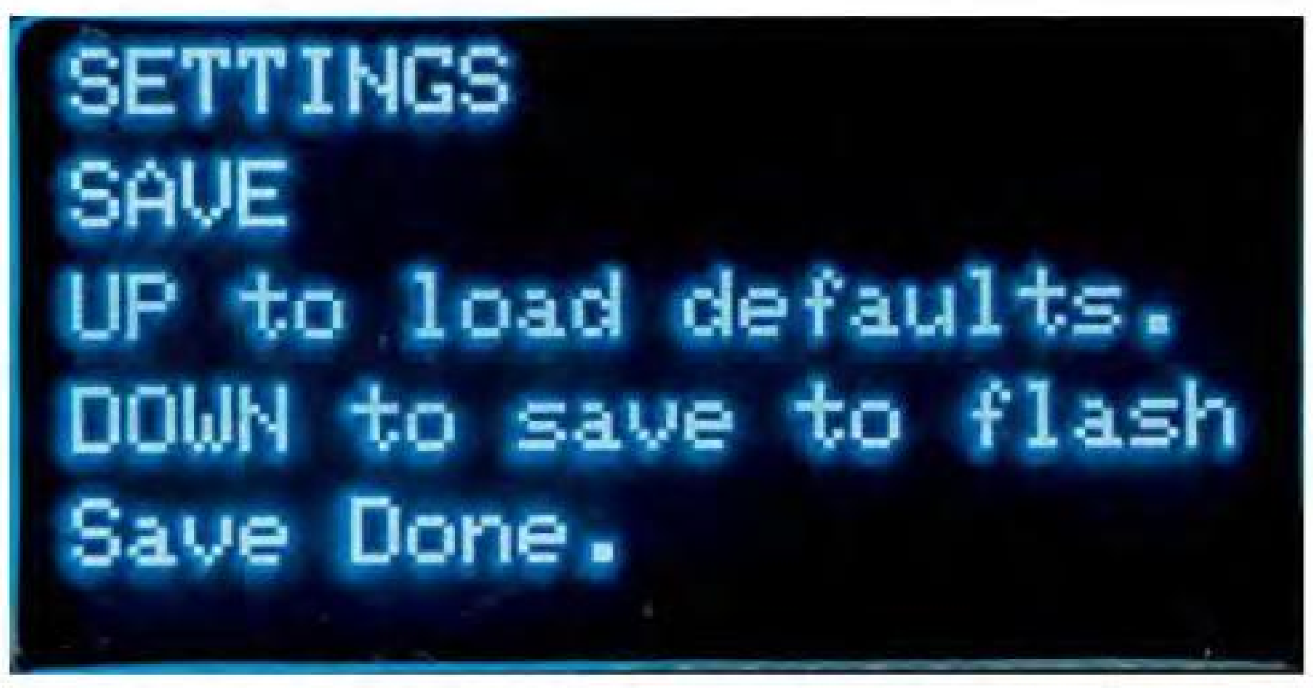
```
SETTINGS
INPUT LEVEL x1 (SW!)
Connect 'OUT' to 'IN'
DOWN to set 0.000891.
Current is 0.000891.
```

Screen 4: connect the input to the output with an appropriate RCA cable for the next steps. Ensure the range switch S6 is set to 1x and press DOWN when prompted, then OK.



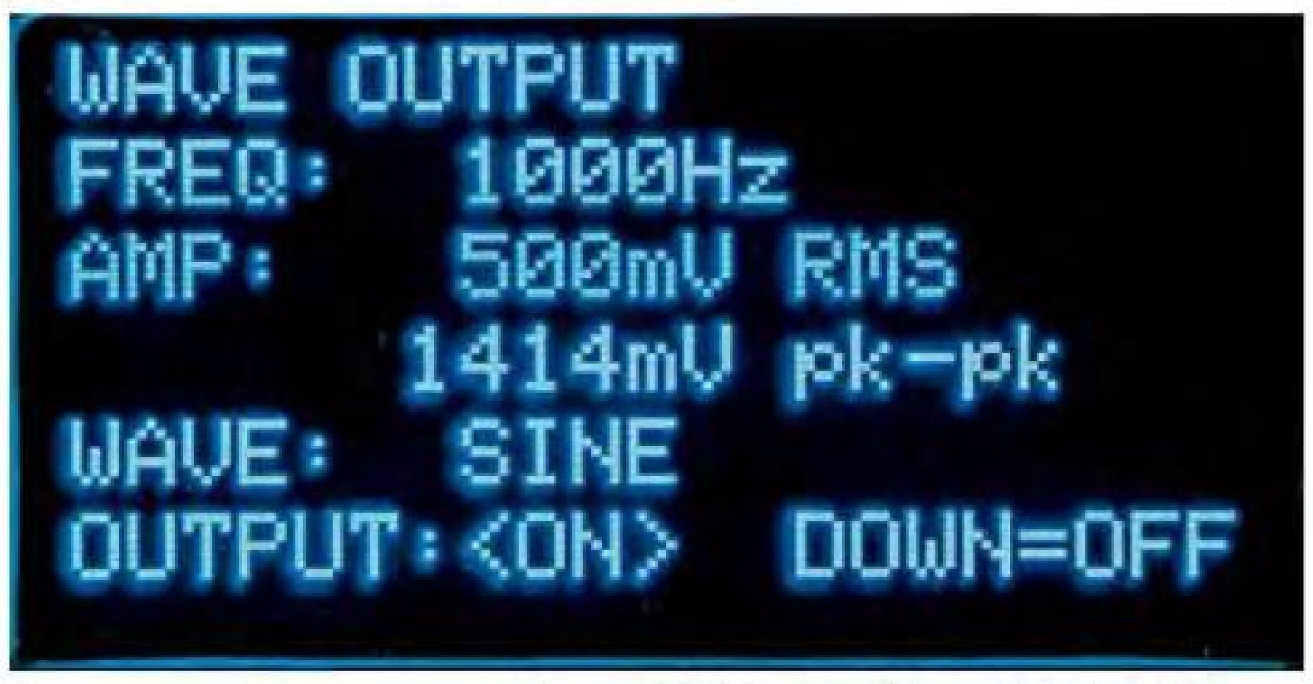
```
SETTINGS
INPUT LEVEL x10 (SW!)
Connect 'OUT' to 'IN'
DOWN to set 0.008290.
Current is 0.008288.
```

Screen 5: follow the prompts and set the switch to 10x. You will see a message if S6 is set to the wrong position or a signal is not detected. Press DOWN to set the scaling factor, followed by OK.



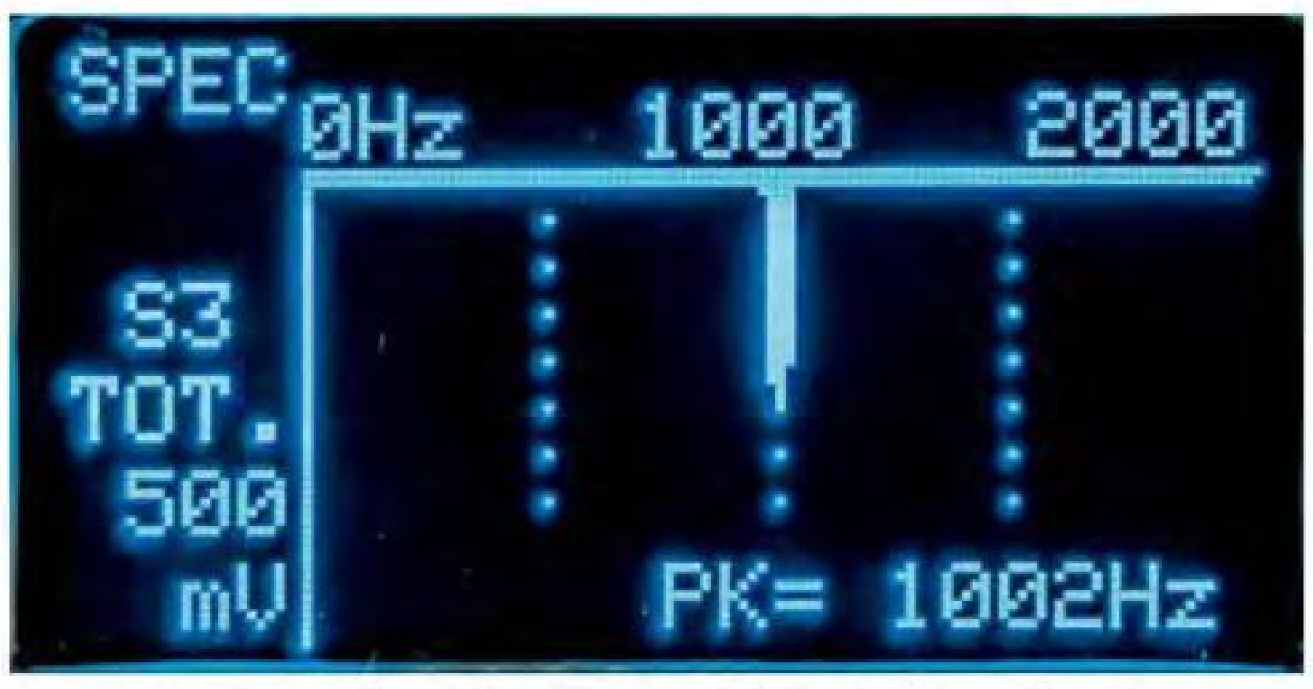
```
SETTINGS
SAVE
UP to load defaults.
DOWN to save to flash
Save Done.
```

Screen 6: press OK again to see this screen and then DOWN to save the calibration values to flash memory. You will see a message confirming that it was done.

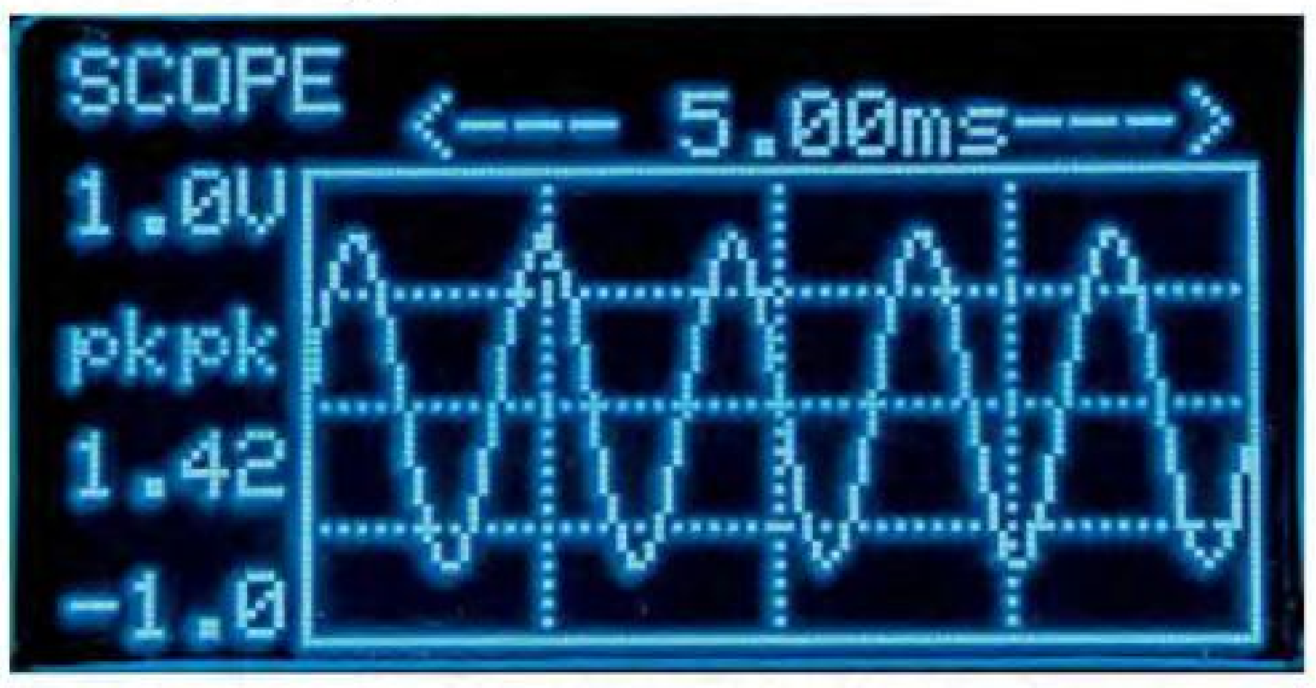


```
WAVE OUTPUT
FREQ:   1000Hz
AMP:    500mV RMS
         1414mV pk-pk
WAVE:   SINE
OUTPUT:<ON>   DOWN=OFF
```

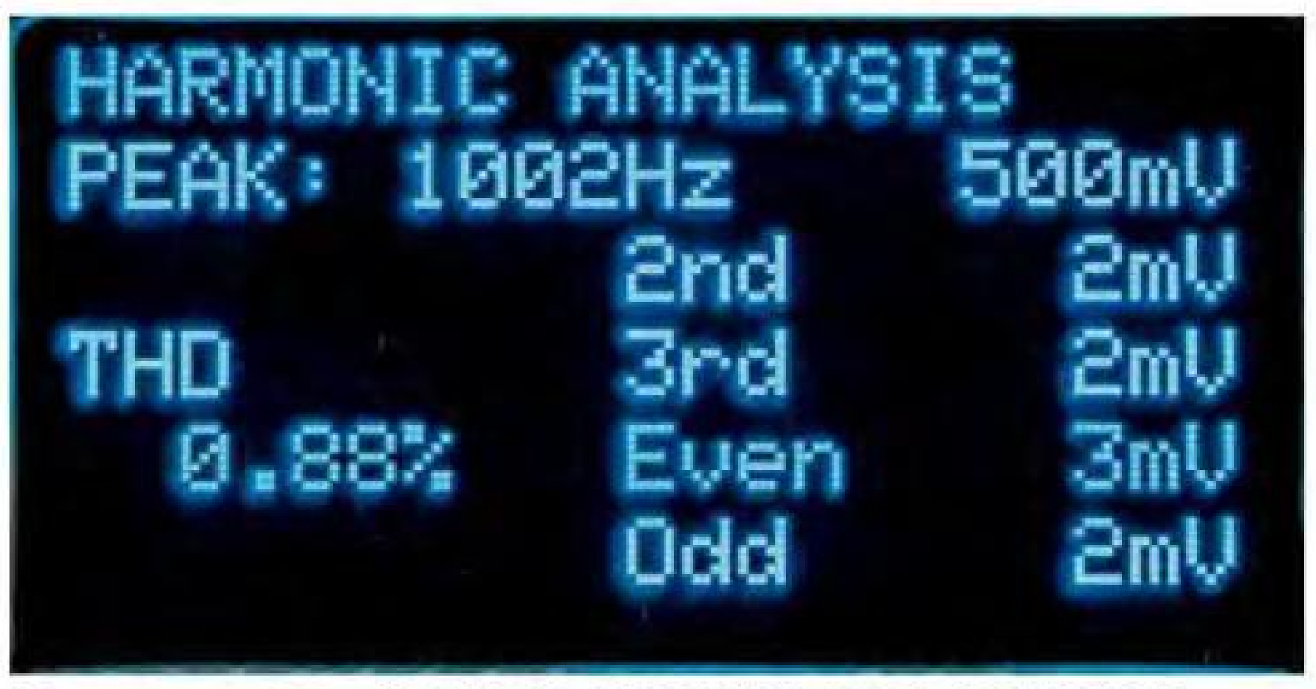
Screen 7: pressing OK on the WAVE OUTPUT screen cycles between the parameters, while UP and DOWN modifies them. The USB serial port can also control the output waveform.



Screen 8: the SPECTRUM display uses UP and DOWN to change the horizontal scaling, while OK toggles the vertical scale between peak and total energy.



Screen 9: the SCOPE display also uses UP and DOWN to change the horizontal scaling. The OK button changes between dot and line displays.



```
HARMONIC ANALYSIS
PEAK: 1002Hz     500mV
            2nd    2mV
THD         3rd    2mV
  0.88%     Even   3mV
            Odd    2mV
```

Screen 10: HARMONIC ANALYSIS provides information about the harmonic content of a waveform. Connecting the input to the output is a good way to check this feature.

If you are measuring the Analyser's output, you will see THD figures around 1% for a sinewave, with about 0.7% due to the output stage and 0.3% due to the input stage. These figures will vary depending on the frequency.

The final mode is the frequency sweep and response. Screen 11 shows the setup, while Screen 12 shows the results. The lower and upper frequencies can be set in powers of 10 between 10Hz and 10kHz, and up to 30 steps can be applied. Each step takes about ⅓ of a second to process.

There is also the option of running a single sweep pass or a continuously updating loop. The default of 10 steps over this range gives a typical display seen in Screen 12; this is with the output connected to the input.

The horizontal frequency scale is logarithmic; the dashed grid lines correspond to the second and fifth divisions of their respective decades. The vertical scale is adjustable with the UP and DOWN buttons and the intermediate grid line corresponds to the -3dB point.

As an exercise, we connected a simple low-pass RC filter circuit (using a 1kΩ resistor and a 1µF capacitor) between the input and output. As expected, the SWEEP showed a -3dB point around 160Hz, rolling off more at higher frequencies.

While a direct connection from output to input should give a perfectly flat response, there are slight dips at 10Hz and 10kHz as the low-pass and high-pass filters start to take effect. The small peak around 20Hz is a side-effect of the windowing function.

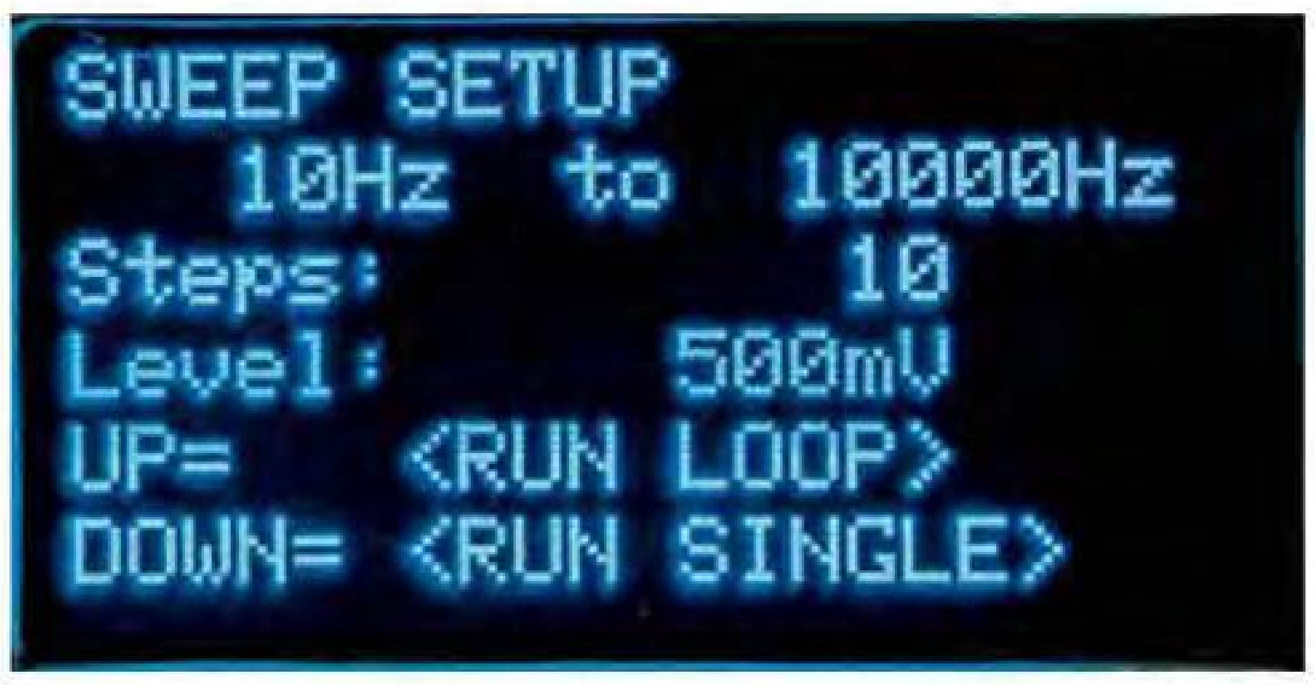
Pressing OK from the graph page will end the looping behaviour, or if <OK> is shown, return to the setup menu.

Computer control

Since we have a USB port on the Pico 2, we use it to provide alternative controls and data outputs. We recommend using a terminal program such as TeraTerm (on Windows) or minicom (on Linux), as the Arduino serial monitor is quite basic.

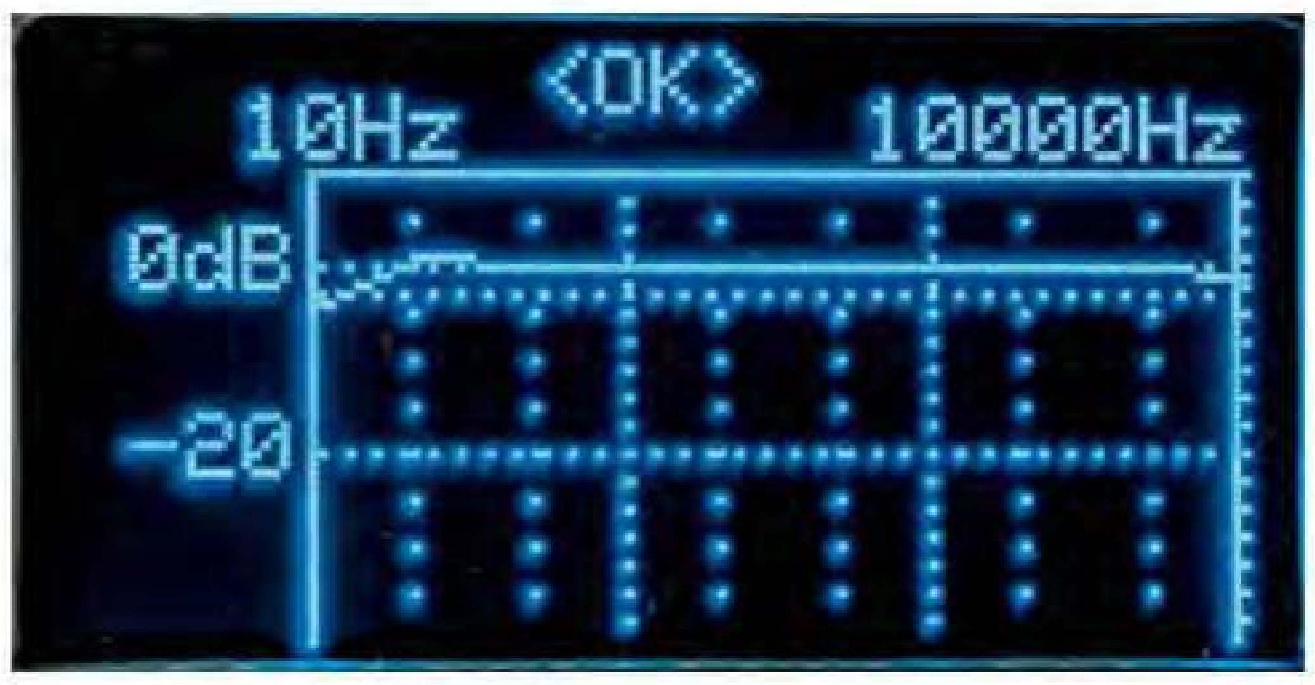
Most commands are followed by the Enter key, but the commands that emulate the buttons on the Analyser act instantly.

For a full list of commands, type "?" and press Enter. The keys listed at the bottom emulate the four onboard buttons.



```
SWEEP SETUP
   10Hz  to   10000Hz
Steps:          10
Level:        500mV
UP=     <RUN LOOP>
DOWN= <RUN SINGLE>
```

Screen 11: SWEEP uses the UP, DOWN and OK buttons like the OUTPUT mode. There is the option of running a single sweep pass or performing a continuous loop.



Screen 12: in this display, the UP and DOWN buttons change the vertical scaling; the unlabelled horizontal line being the -3dB point compared to the set level at the output.

Most of the remaining commands emulate the controls of the WAVE OUTPUT mode. Since they will work while another mode is active, they can save you the time of cycling between modes to change settings and then trying to view the results.

"a" or "A" followed by a number will set the output RMS amplitude in millivolts. For example, "a500" sets the output to 500mV RMS. Similarly, "p" or "P" will set the peak-to-peak amplitude in millivolts. The "f"/"F" option sets the frequency in Hertz, the "w"/"W" command sets the type of waveform, while "o"/"O" turns the wave output off or on.

Note that setting parameters too high might result in corrupted waveforms.

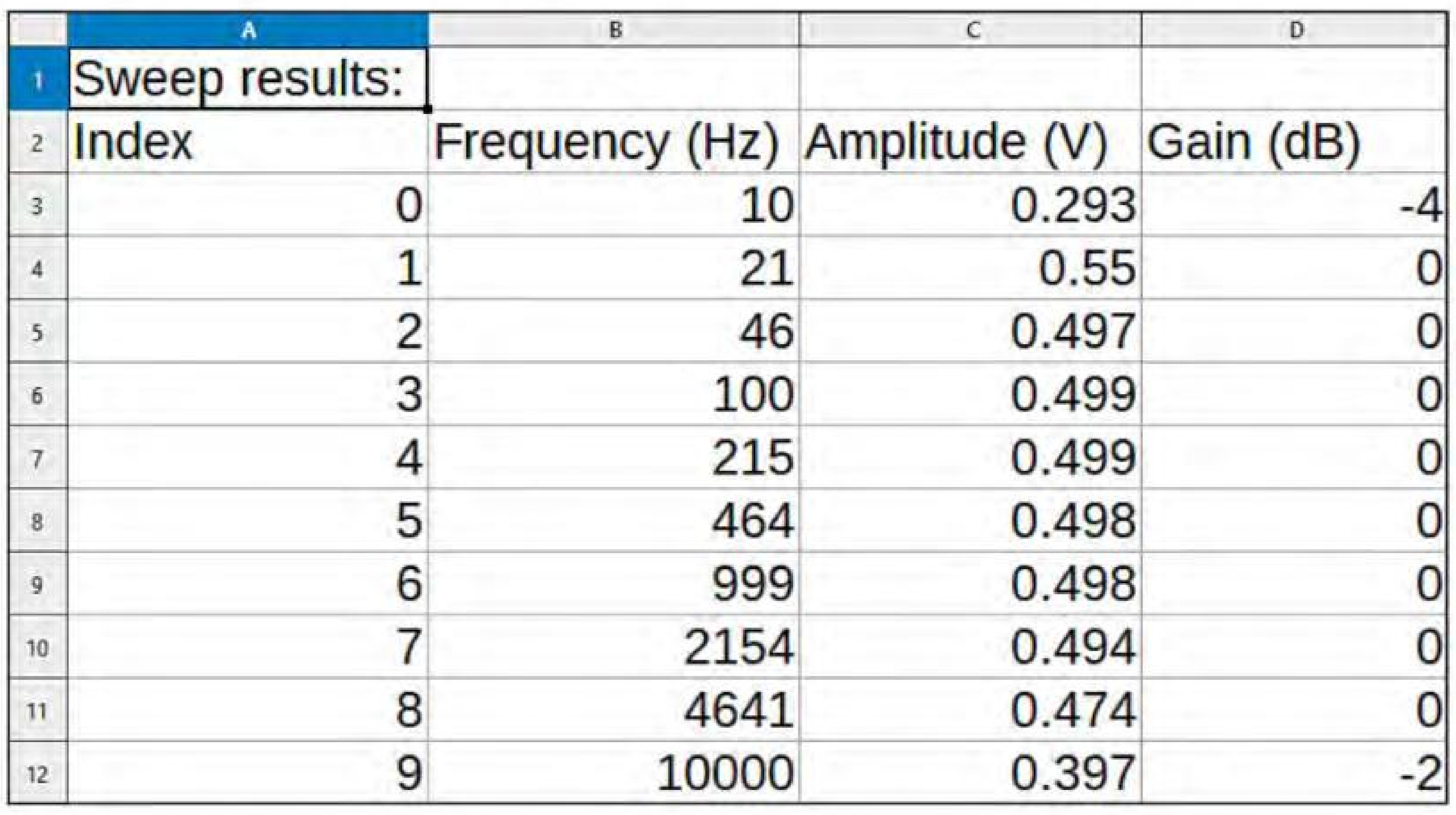
Another command, "d"/"D", provides a 'data dump' of the next scan in the SCOPE, SPECTRUM, HARMONIC ANALYSIS or SWEEP modes. The data is formatted similarly to a CSV (comma-separated variable) file, so you can paste the data directly into spreadsheet programs that support CSV data.

For a SWEEP, the dump will occur after the next pass has completed; Screen 13 shows the same data as in Screen 12 as a spreadsheet.

Finally, the "~" command resets the Pico 2. Holding the BOOTSEL button while issuing this command will enter bootloader mode for reprogramming the Pico.

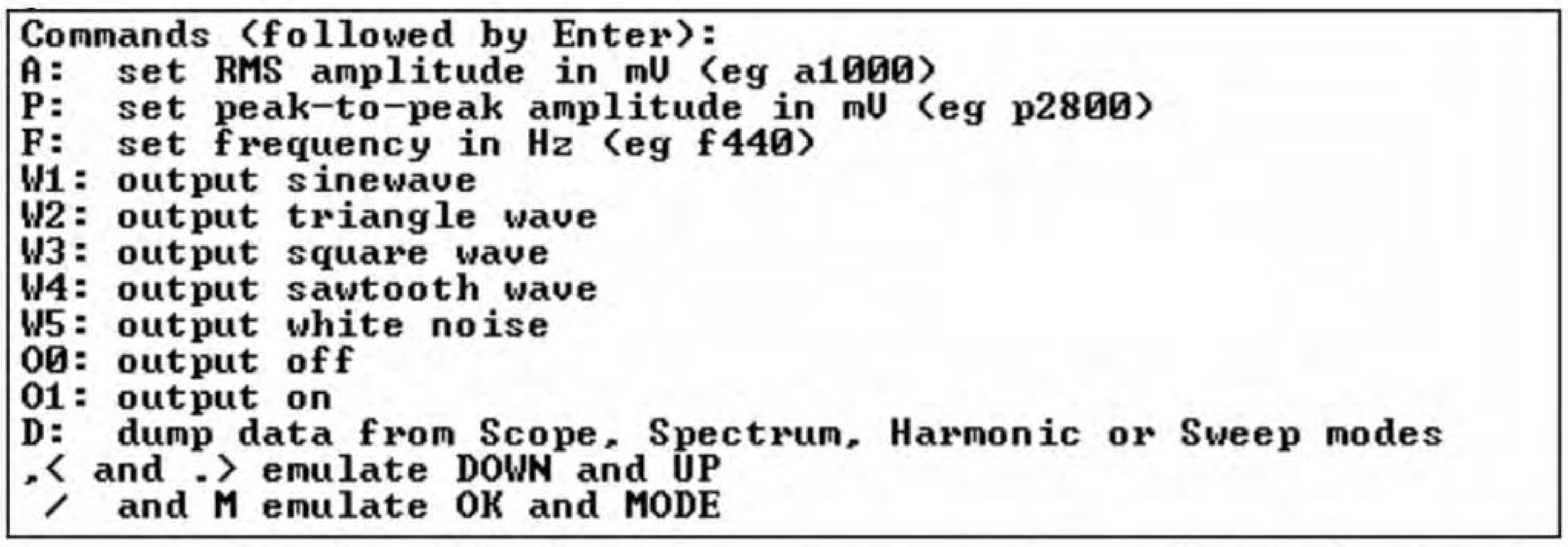
Conclusion

The Pico 2 Audio Analyser is a simple and compact device that uses little in the way of hardware apart from the Pico itself. Its performance is modest, but we think its simplicity and cost make it a handy tool. **PE**



	A	B	C	D
1	Sweep results:			
2	Index	Frequency (Hz)	Amplitude (V)	Gain (dB)
3	0	10	0.293	-4
4	1	21	0.55	0
5	2	46	0.497	0
6	3	100	0.499	0
7	4	215	0.499	0
8	5	464	0.498	0
9	6	999	0.498	0
10	7	2154	0.494	0
11	8	4641	0.474	0
12	9	10000	0.397	-2

Screen 13: the "d" command at the serial terminal triggers a dump of data in CSV format. We pasted the data shown here, from the SWEEP mode, into a spreadsheet program.



```
Commands (followed by Enter):
A:  set RMS amplitude in mV (eg a1000)
P:  set peak-to-peak amplitude in mV (eg p2800)
F:  set frequency in Hz (eg f440)
W1: output sinewave
W2: output triangle wave
W3: output square wave
W4: output sawtooth wave
W5: output white noise
O0: output off
O1: output on
D:  dump data from Scope, Spectrum, Harmonic or Sweep modes
,< and .> emulate DOWN and UP
 /  and M emulate OK and MODE
```

Screen 14: this view of TeraTerm shows the commands provided by the virtual USB serial port. The list can be shown by using the "?" command.

AUDIO OUT

By Jake Rothman

Absolutely phased out (all-pass filters, part one)

I was listening to 10cc's studio masterpiece from 1975, *I'm Not in Love*, and the line "It's just a silly phase I'm going through" made me think it's about time I clarified all this 'phase business' that pervades audio.

There's too much ambiguity about the word phase, which is often conflated with polarity, which is about positive and negative parts of the waveform. An inverter or swapping the wires in a balanced line or loudspeaker feed can be used to reverse polarity, making the waveform upside-down, as shown in Fig.1.

Unfortunately, many people incorrectly call signal inversion "flipping the phase", "reversing the phase" or "out of phase". Absolute polarity, also sometimes called absolute phase, is a voltage or pressure effect where a positive-going signal event comes out as a positive-going voltage.

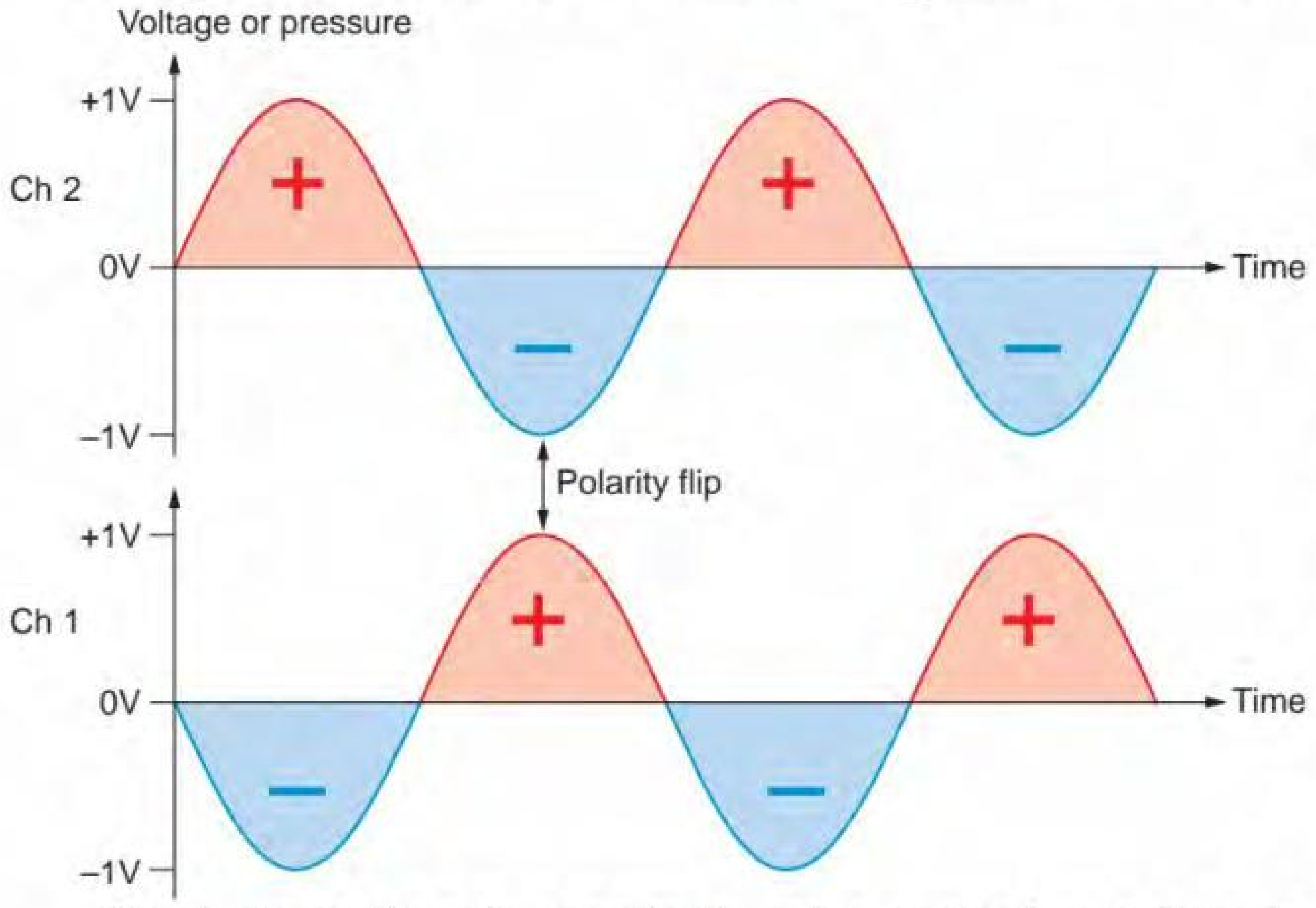

Voltage or pressure
+1V
Ch 2
+
+
0V
Time
−
−
−1V
Polarity flip
+1V
Ch 1
+
+
0V
Time
−
−
−1V

Fig.1: inverting the polarity of a waveform flips it about the time axis.

Phase shift is really a function of time, a time delay that shifts waveforms relative to each other, as illustrated in Fig.2. It can also shift the relative cycles of different frequencies, such as harmonics, often making the waveform more symmetrical, as shown in Fig.3.

In audio, we should use phase to denote only a time-based alignment or delay, but it is notoriously difficult to shift culturally ingrained linguistic errors. I was once told to relabel a button on an active crossover from "invert" to "phase Ø" on the basis it was what the customer expected.

The Latin capital letter O or Ø is often used to denote invert on mixer channels (Photo 1) because there is little panel space, which I think is acceptable.

Editor's note: delaying a sinewave (or any waveform symmetrical about 0V) by half a cycle has the same effect on the shape as inverting it, hence voltage inversion being referred to as 180° phase shift or inversion. However, that does not apply for asymmetric waveforms.

Also, besides 180° and integer multiples of it, all other phase shifts can only be reliably created by delaying the signal (or by frequency-dependent networks that are equivalent to delays), as implied above.

Get it right

It is the duty of the audio engineer to ensure his equipment does not invert the polarity, since this can cause problems later on. This is important because many signal processing circuits necessarily invert the voltage, which can cause cancellations when effects are combined in the studio.

The famous Quad 303 and 405 amplifiers inverted the signal. Many CD players were also inverting because the R/2R digital-to-analog converter (DAC) fed an inverting amplifier. There is even a polarity-corrected CD available of XTC's *Skylarking* album because it was thought this was why the first CD release didn't sound as good as the original vinyl.

Photo 1: the polarity inversion button on a mixer channel strip.

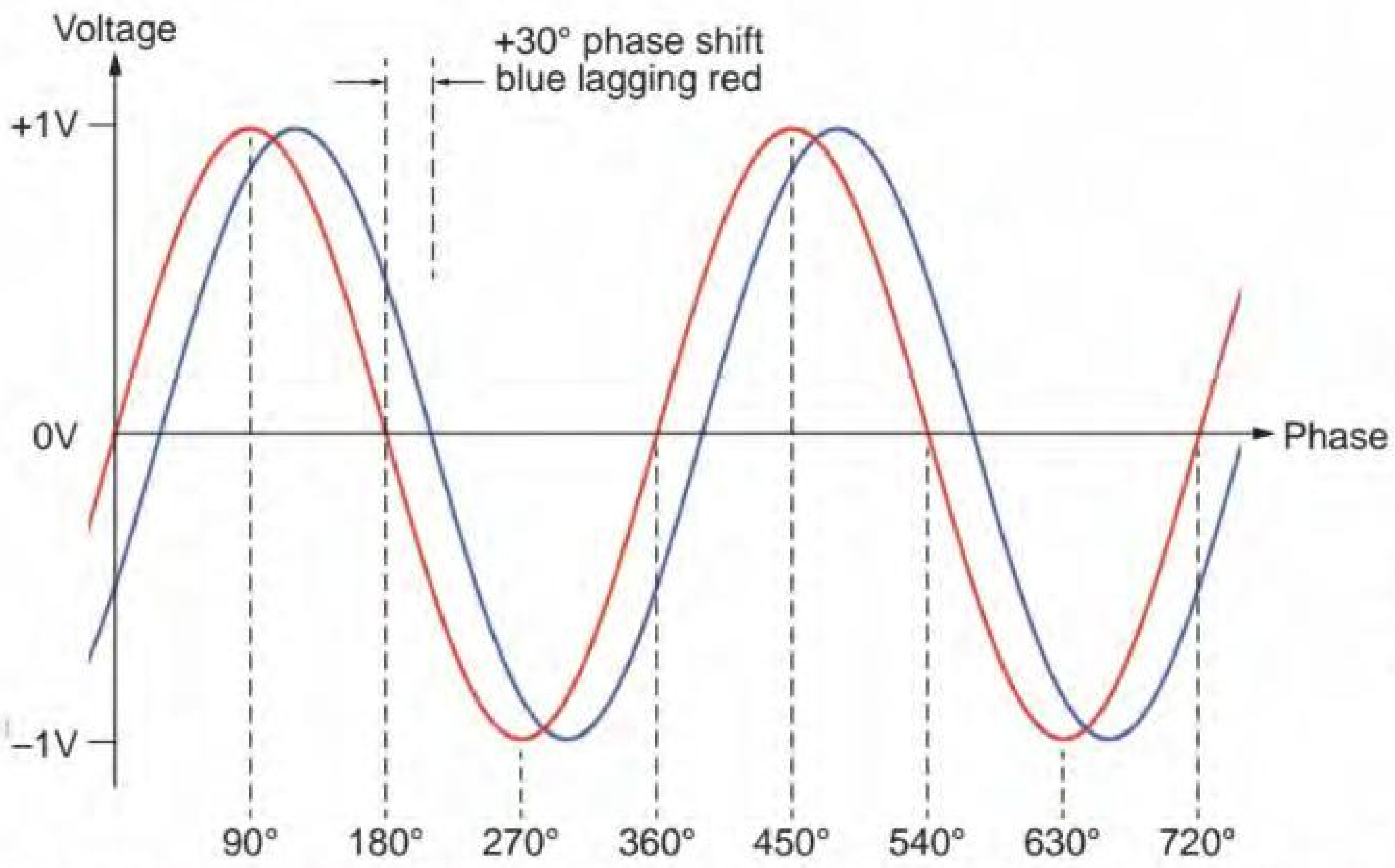

Voltage
+30° phase shift
blue lagging red
+1V
0V
Phase
−1V
90°
180°
270°
360°
450°
540°
630°
720°

Fig.2: phase shift displaces the waveform in time.

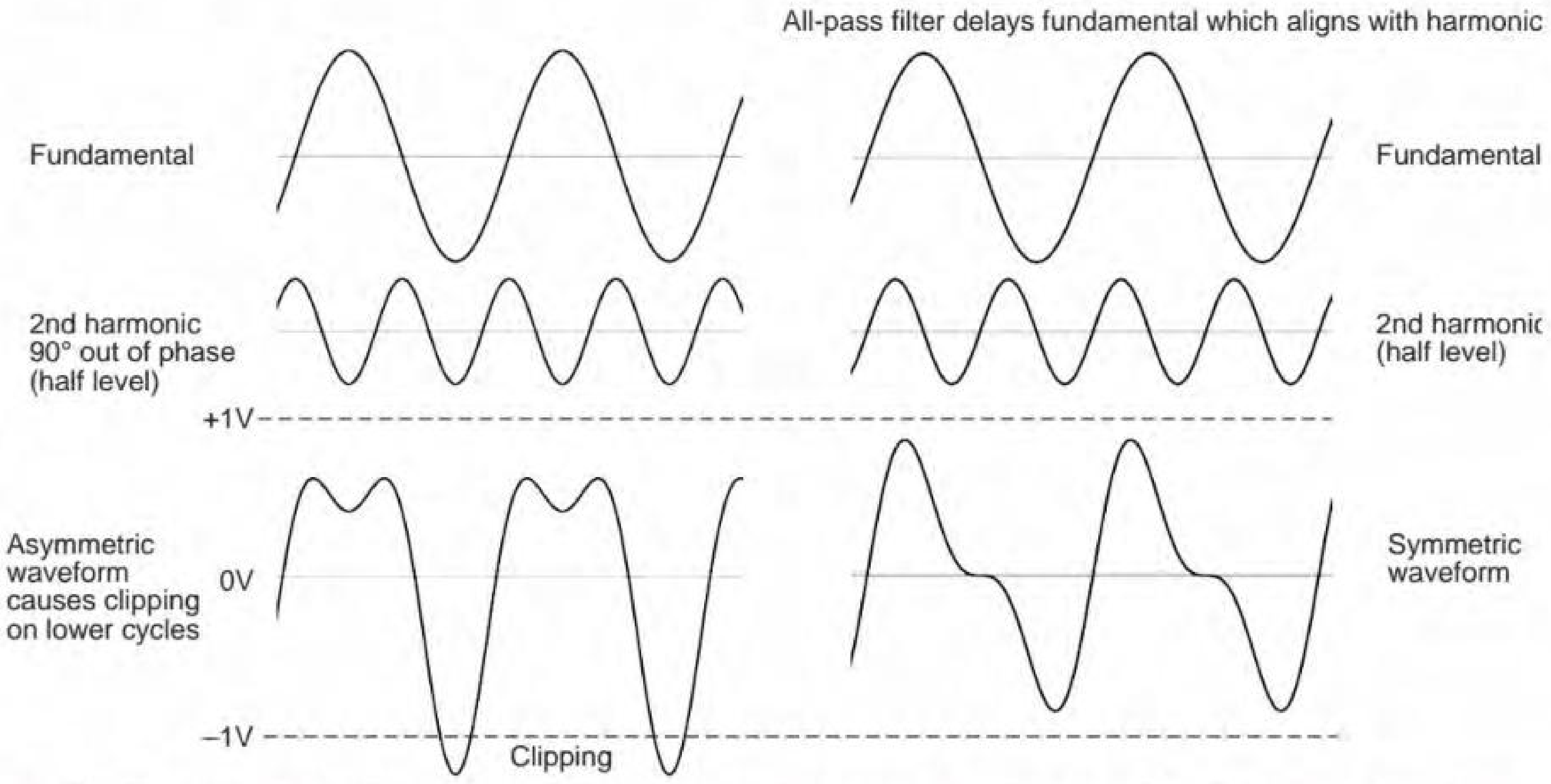

All-pass filter delays fundamental which aligns with harmonic
Fundamental
Fundamental
2nd harmonic
90° out of phase
(half level)
2nd harmonic
(half level)
+1V
Asymmetric
waveform
causes clipping
on lower cycles
0V
Symmetric
waveform
−1V
Clipping

Fig.3: frequency-dependent phase shift can change the waveshape.

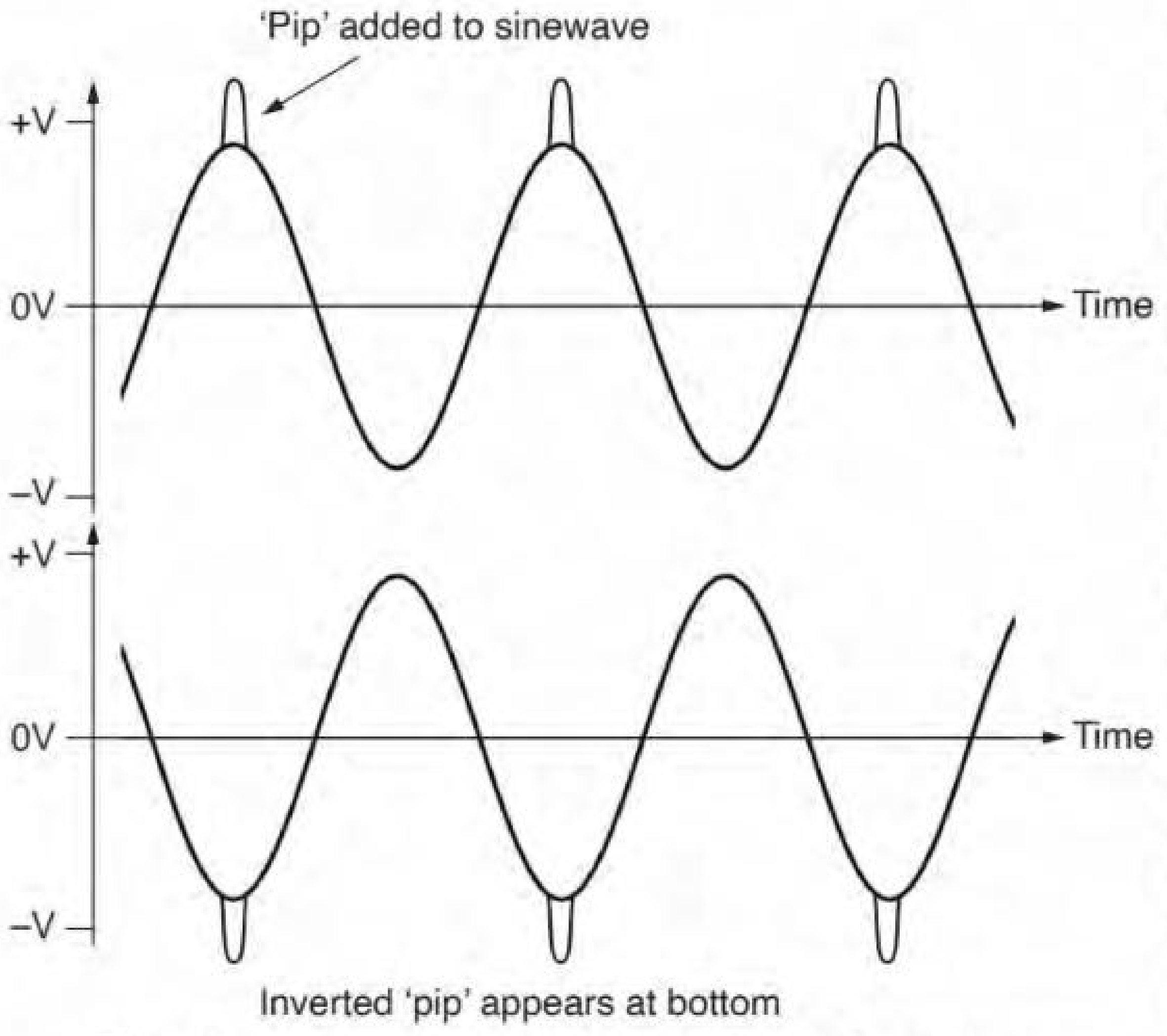


Fig.4: asymmetric test waveforms are useful for detecting polarity reversals and phase shifts.

Vinyl playback systems usually have correct polarity. People seem to vary in their ability to detect absolute polarity. Personally, if I polarity-flip most music, I cannot hear a difference, but I feel reassured if it's right. This can be easily checked with a scope or digital audio workstation (DAW) using solo instruments with positively biased waveforms such as kick drums and trumpets.

Ted Fletcher of tfpro, an audio engineer from whom I have learned a lot, is convinced kick drums sound better if the absolute polarity is correct. Ted sadly died on the 17th of August 2025 at age 87, and yes, I did think about interviewing him!

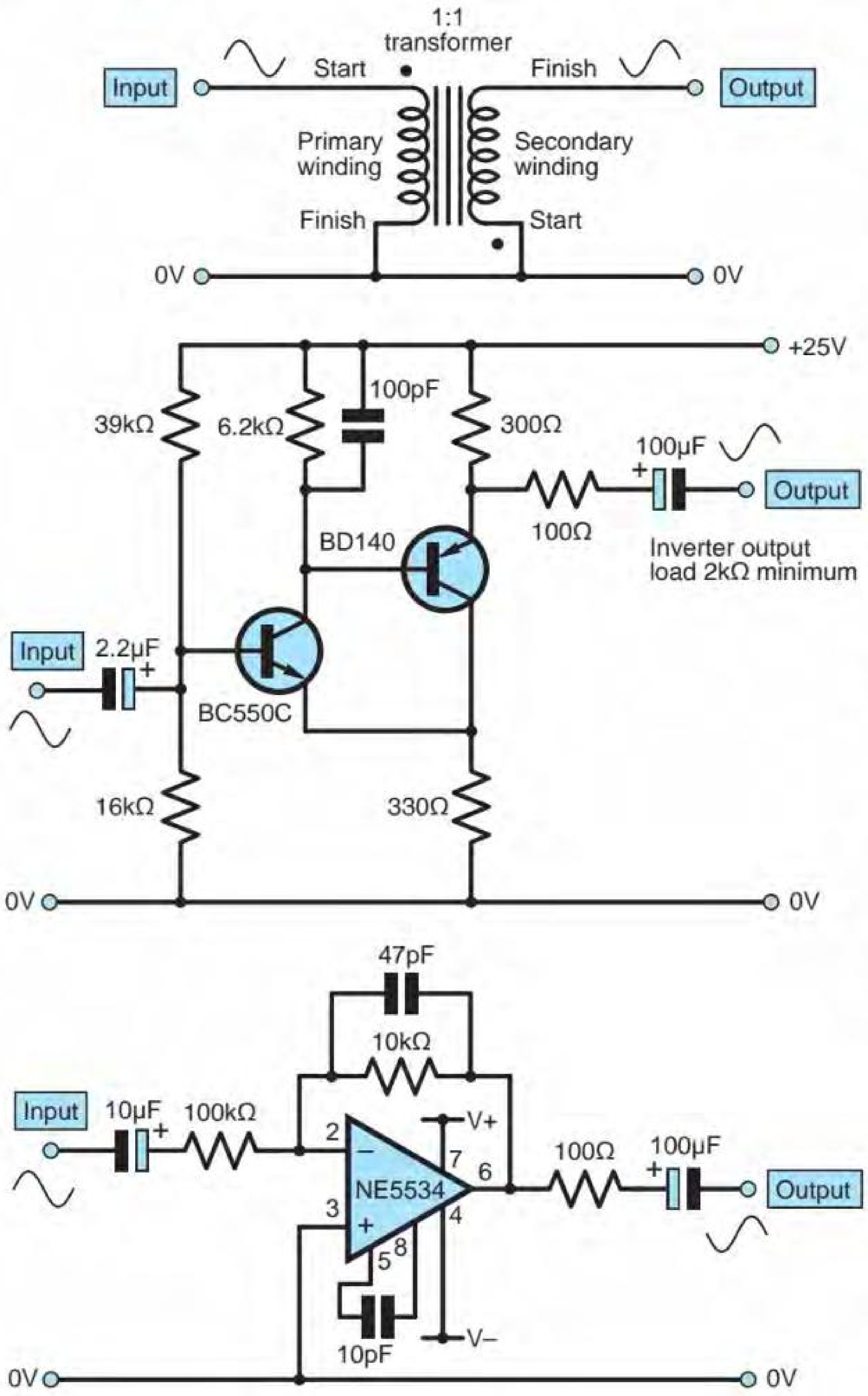


Fig.5: classic audio inverter circuits: transformer based, common-emitter and op amp versions.

However, with complex multi-track mixes, the polarity is likely to be pretty random. In fact, a good mixing engineer will ensure it is as random as possible to get the loudest mix. Having occasional coincident peaks or an asymmetrical mix with more energy on the positive or negative side wastes headroom.

Australian mixing engineer Michael Paul Stavrou, in his book *Mixing With Your Mind*, described this approach as "maximum illusion/minimum voltage". He would change the polarity of some instruments (p193) to prevent peaks building up in the mix, for instance, making the bass line the opposite polarity to the kick drum.

When listening to one's own voice in headphones while singing in the studio or broadcasting, it's essential that the absolute polarity is correct. Very strange effects can occur if cancellation occurs with bone conduction through one's head or acoustic leakage through the ear pads.

It can be difficult to check the polarity of a simple sinewave through a circuit, unless one uses a dual-trace oscilloscope. One trace needs to be connected to the output of the signal generator and triggered from it, the other to the probe.

Another way to do it with a single-trace scope is to use an asymmetrical test waveform, such as the ones illustrated in Fig.4. A distorted sinewave with even harmonics should do it, such as a full-wave rectified one. These waveforms are also good for showing frequency-dependent phase shifts.

Photo 2: checking speaker polarity using a battery.

With loudspeakers, it's simply a matter of connecting a low-voltage (3-6V) battery and seeing if the diaphragm moves out, as shown in Photo 2. If the cone jumps outwards when the battery is connected, the terminal going to the positive on the battery is marked "+" or colour-coded red.

The positive connection denotes positive polarity. With microphones, a gentle puff of air like saying a plosive word, such as "pop" or "bus", should give a positive-going voltage for the positive polarity condition.

One engineer's polarity test was to burst a crisp packet right near my ear while I was miking up the drum kit. Very annoying!

It's a rare thing in life where two wrongs make a right, but if the polarity is wrong, it can be fixed by an inversion somewhere else in the chain. In a stereo system, it's fine if both speakers are wired with the wrong polarity, but a horrid hole-in-the-middle effect and bass loss is the result if only one side is inverted.

This used to be one of the biggest problems when I sold Hi-Fi systems. It didn't help that the speaker twin cable we supplied was clear PVC insulated, printed with very pale "+" signs along one side that nobody over 40 could see.

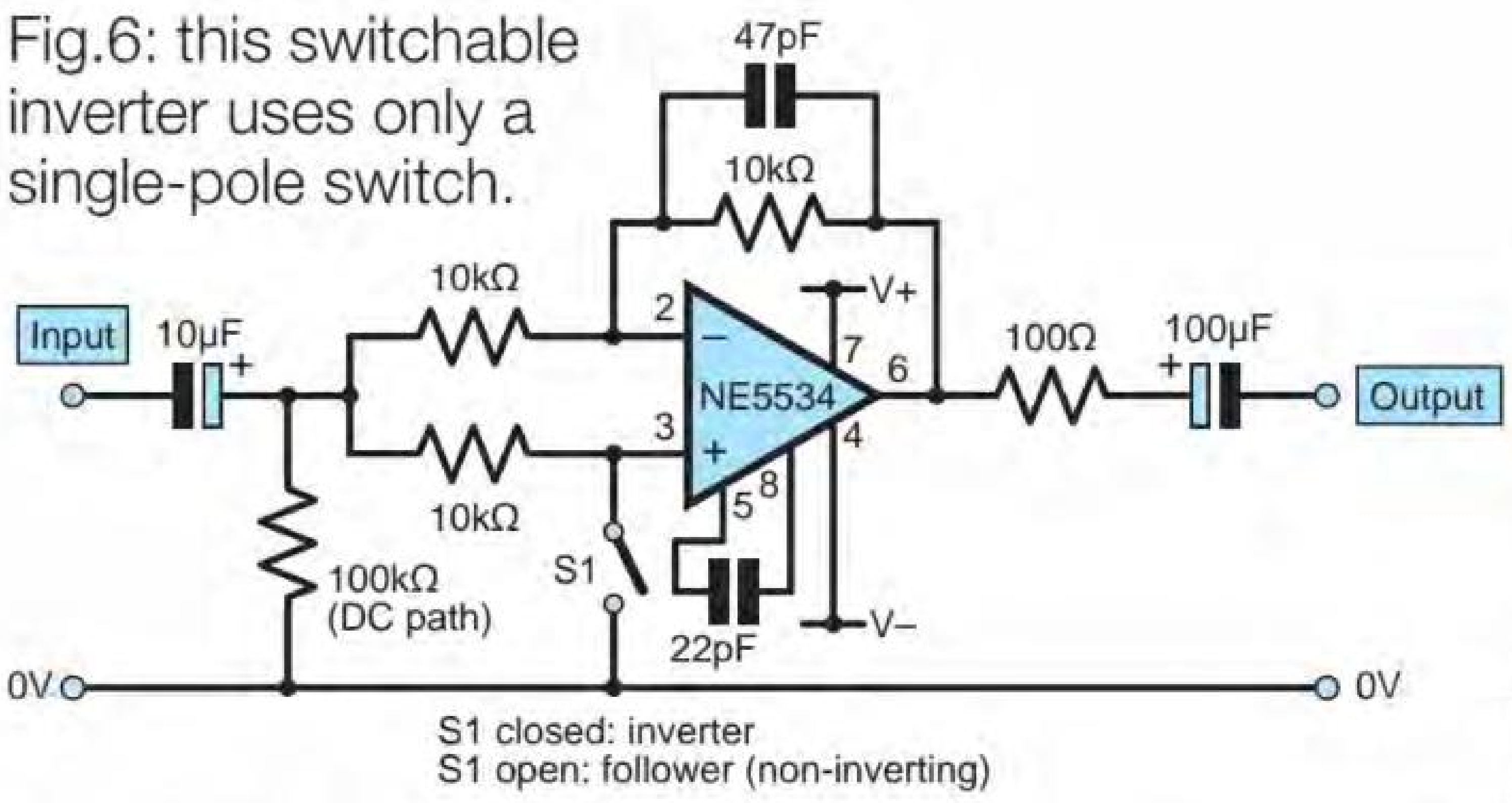
The inverter

The simplest audio inverter is a transformer, but it's cheaper (and introduces less distortion) to do it with an op amp, as shown in Fig.5.

A switchable inverter can be made by using a switch to either feed the signal through or bypass an inverter, but there is a much better circuit that only uses a single-pole switch, shown in Fig.6. I suspect this evolved from differential op amp circuits – I don't know who invented it.

With the switch in the open position, it is a buffer/follower, and when the switch is closed, it becomes an inverter, with the resistor in parallel with the input.

Fig.6: this switchable inverter uses only a single-pole switch.



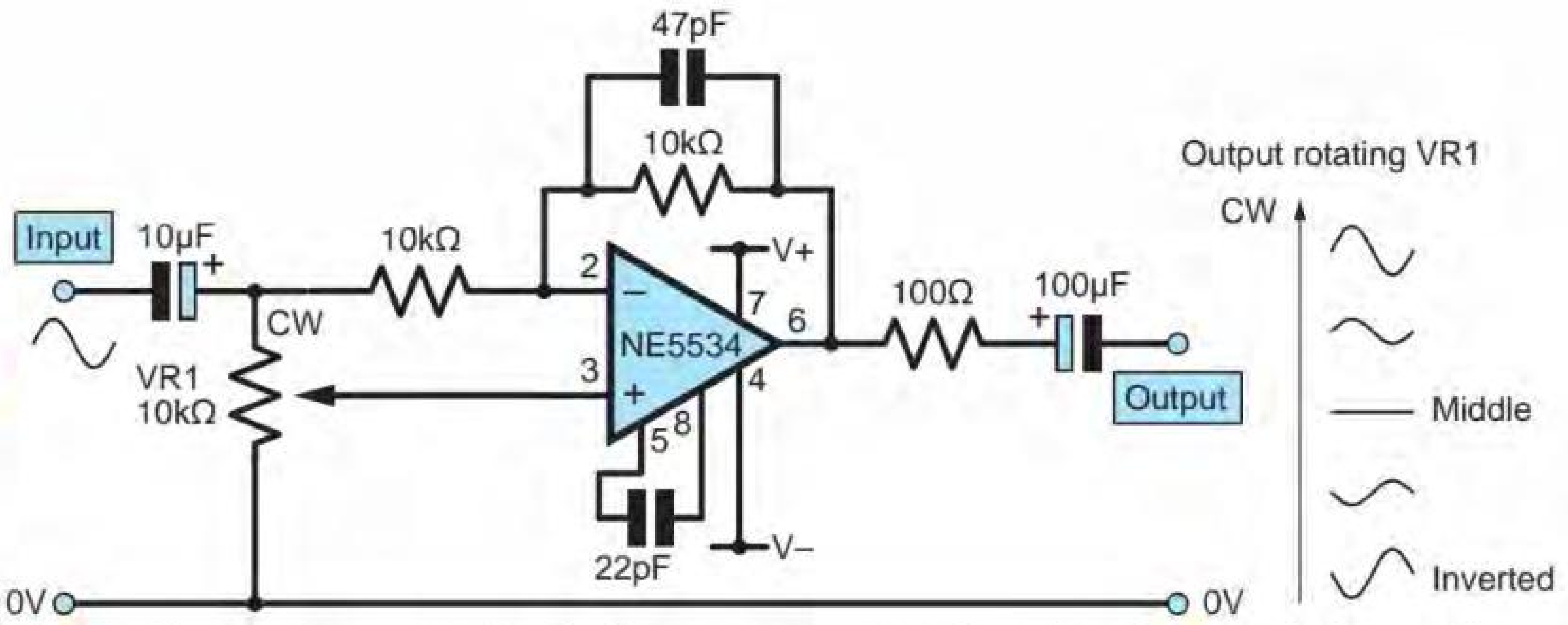


Fig.7: the 'attenuverter' gives a range of gains from -1 (inverting amplifier) through zero to +1 (unity-gain buffer).

The 'attenuverter'

I thought I knew every analog audio circuit block in existence, but Grindle brought the term 'attenuverter' to my attention. It is basically a level control (shown in Fig.7) that has zero output in the middle of its rotation, gives an inverted output on the anti-clockwise side and an uninverted output when turned clockwise.

It is very useful for mixing two things together and for synth control voltages. The first published attenuverter circuit I could find was by Ben Sullivan in the Circuit Ideas section of *Electronics and Wireless World*, August 1994 (p688).

Phase vs waveform shape

Like polarity, phase is mostly discernible in a relative context. You can easily hear when one speaker in a stereo pair is inverted, but if both are inverted it may go unnoticed in perpetuity. With phase, a single frequency (a sinewave) at any phase has an identical waveform.

Take a harmonically complex wave and change the phase of those harmonics relative to each other, and the overall shape may change dramatically, but it will sound the same unless the phase shift is extreme and changing. I've made a video on this. It is the frequency amplitude detection model of human hearing, which works like a spectrum analyser, that's responsible for this psychoacoustic effect.

Phase errors

If different sources are combined, such as in multi-way speakers, correct phase becomes critical. Otherwise, unwanted cancellation notches occur in the frequency response. The main causes of phase shift in audio are filters and time delays, both acoustic and those from digital processing (latency).

It was believed until the mid-1980s that phase shifts were generally inaudible, but improved source quality has enabled the effect of scrambled phase to be heard. This is a subtle effect, though; it is not as noticeable as frequency response errors. It generally manifests itself as a smearing of transients and a lack of clarity or mushiness.

Timing is also affected, with parts of the frequency spectrum sounding detached. It appears the ear is more sensitive to phase errors at the lower end, since the time shifts are longer. Phase is measured in cycles, so longer cycles have a longer period for the phase shifts to be relative to, giving a more conspicuous effect.

This is often heard on the bass where subwoofers with high-order roll-offs and crossovers are used. At the higher frequency end, it's more subtle, mainly noticeable with MP3 compression (although not as much as the bandwidth limitation), cassette tapes and old CD players and DACs with steep output recovery filters.

Group delay

The cause of these audible phase errors is a variation in delay with frequency. This problem is often described by sound engineers as group delay, but the term group delay is strictly a time delay. Mathematically, it is derived from the rate of change of phase with respect to frequency.

In audio systems, group delay often varies with frequency, so that a 'group' of different frequency signals will arrive at different times to each other. This variation of group delay with frequency is a type of time distortion, with an ideal audio system having a constant, short delay across the whole range.

Rapidly increasing group delay as the frequency decreases, as exhibited by bass reflex enclosures and multiple coupling capacitors, is especially noticeable, manifesting itself as booming and lagging bass.

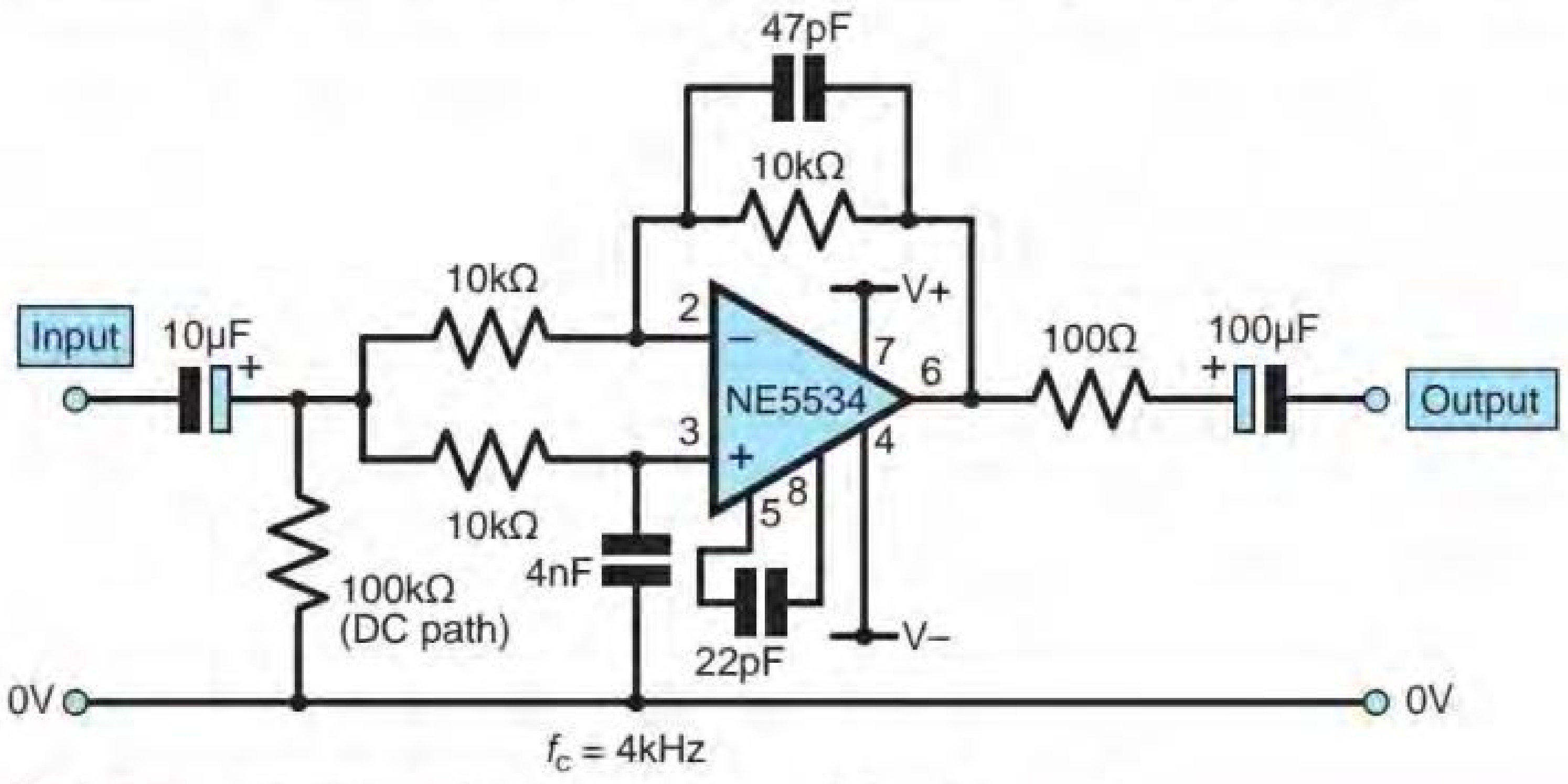


Fig.8: a single-stage all-pass filter circuit.

The all-pass filter

Finally, we get onto the all-singing, all-dancing circuit that can shift phase while having a flat frequency response. It is called an all-pass filter for this reason. It is surprisingly simple to make one by swapping the switch in the switchable inverter for a capacitor, as shown in Fig.8.

If one thinks of a capacitor as a short circuit at high frequencies and an open circuit at low frequencies, it can be seen how the circuit transitions from non-inverting to inverting as the frequency increases, as demonstrated in Fig.9.

The interesting bit is in the middle, where the slope of the phase change is most rapid at 90° where the resistor and capacitor's reactances are equal. This is the turnover frequency, and it is given by the standard formula $1 \div 2\pi RC$. The phase θ (Greek letter theta) is measured in degrees and is given by $2\tan^{-1}(1 \div 2\pi RC)$; it is 90° at this point.

The time delay is given by $t = 2RC$. I expected this 90° point to be where the group delay is highest in accordance with the maths, but

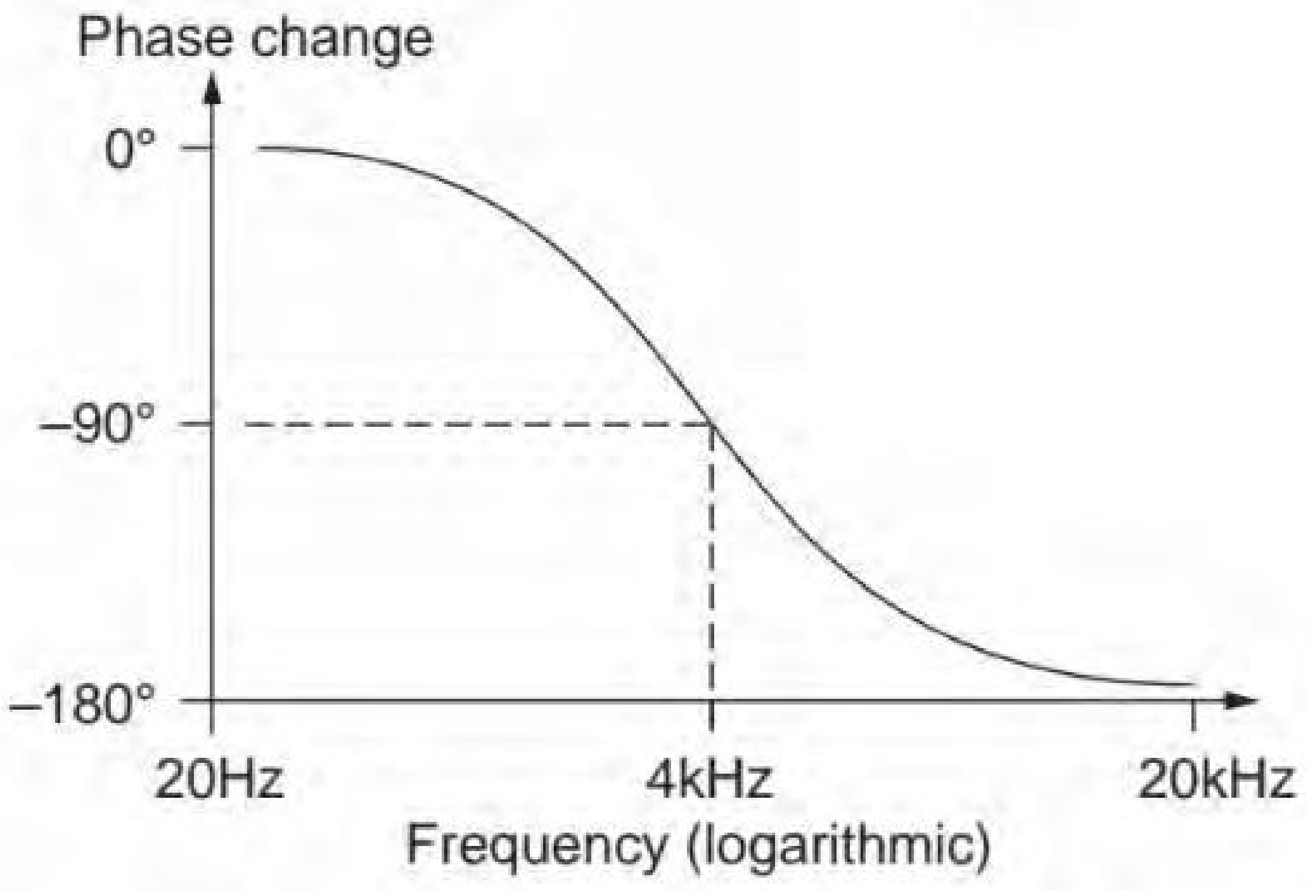


Fig.9: the polarity/phase change of the all-pass filter in Fig.8 vs frequency.

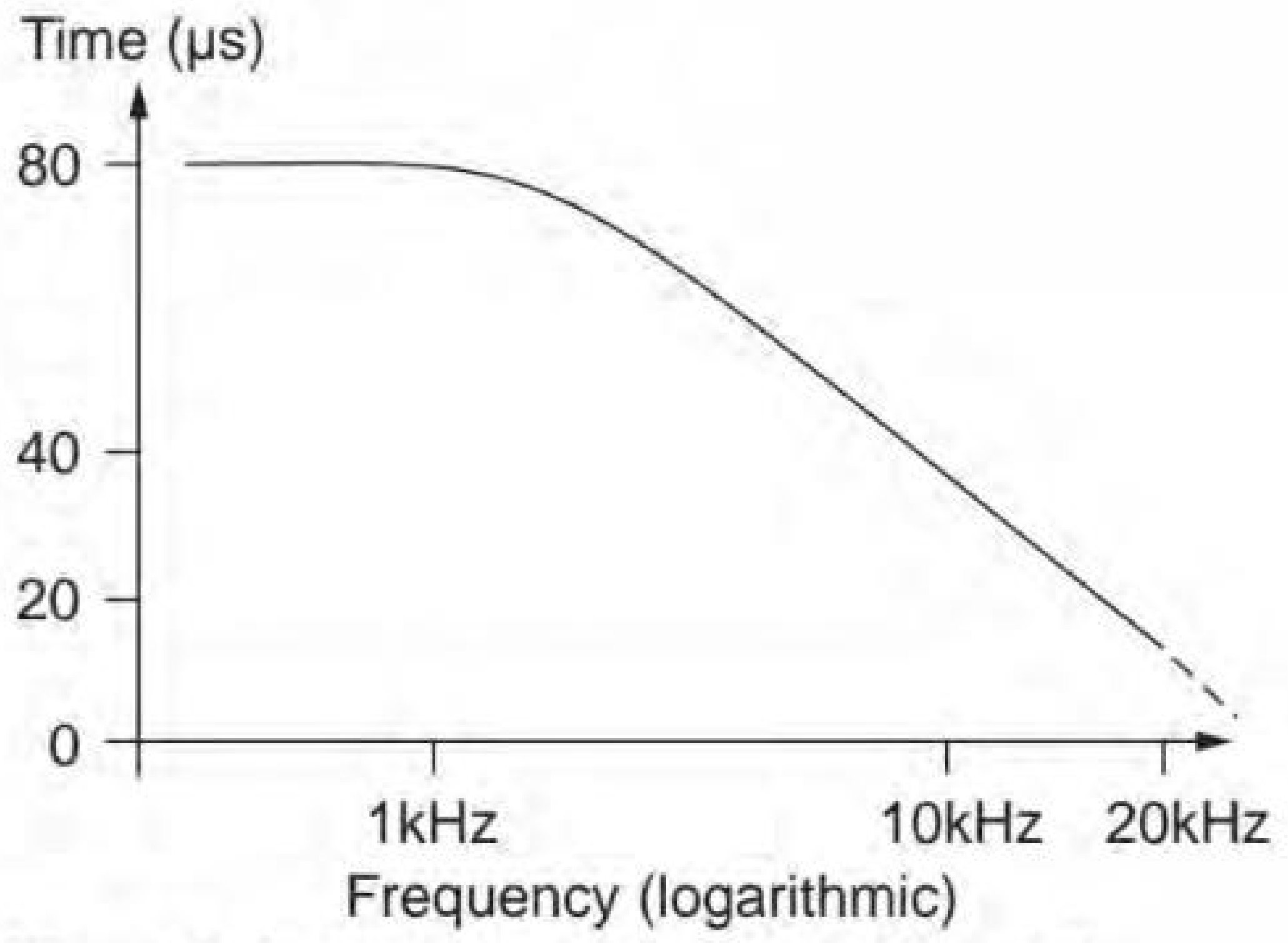


Fig.10: the group delay of the all-pass filter in Fig.8.

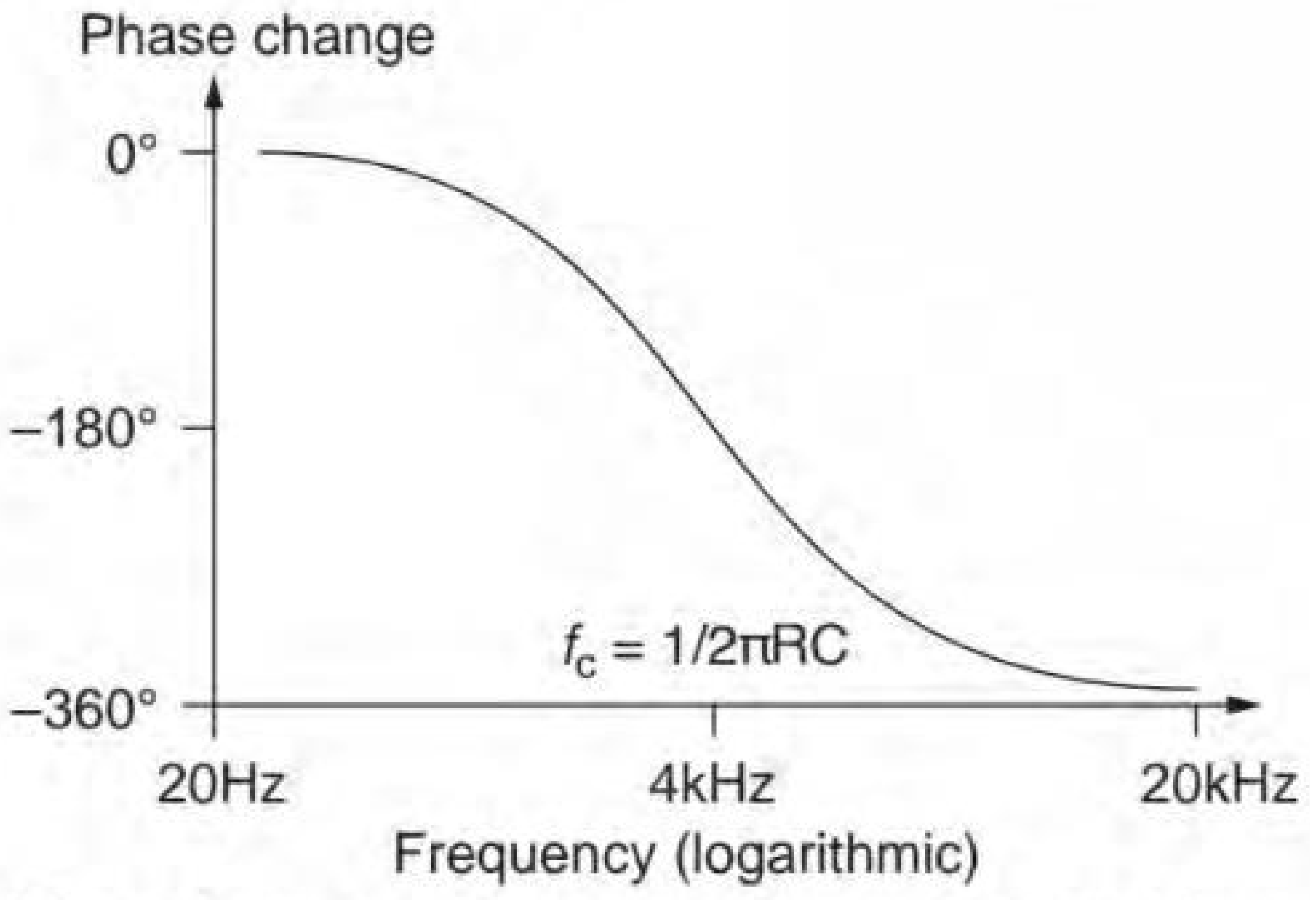


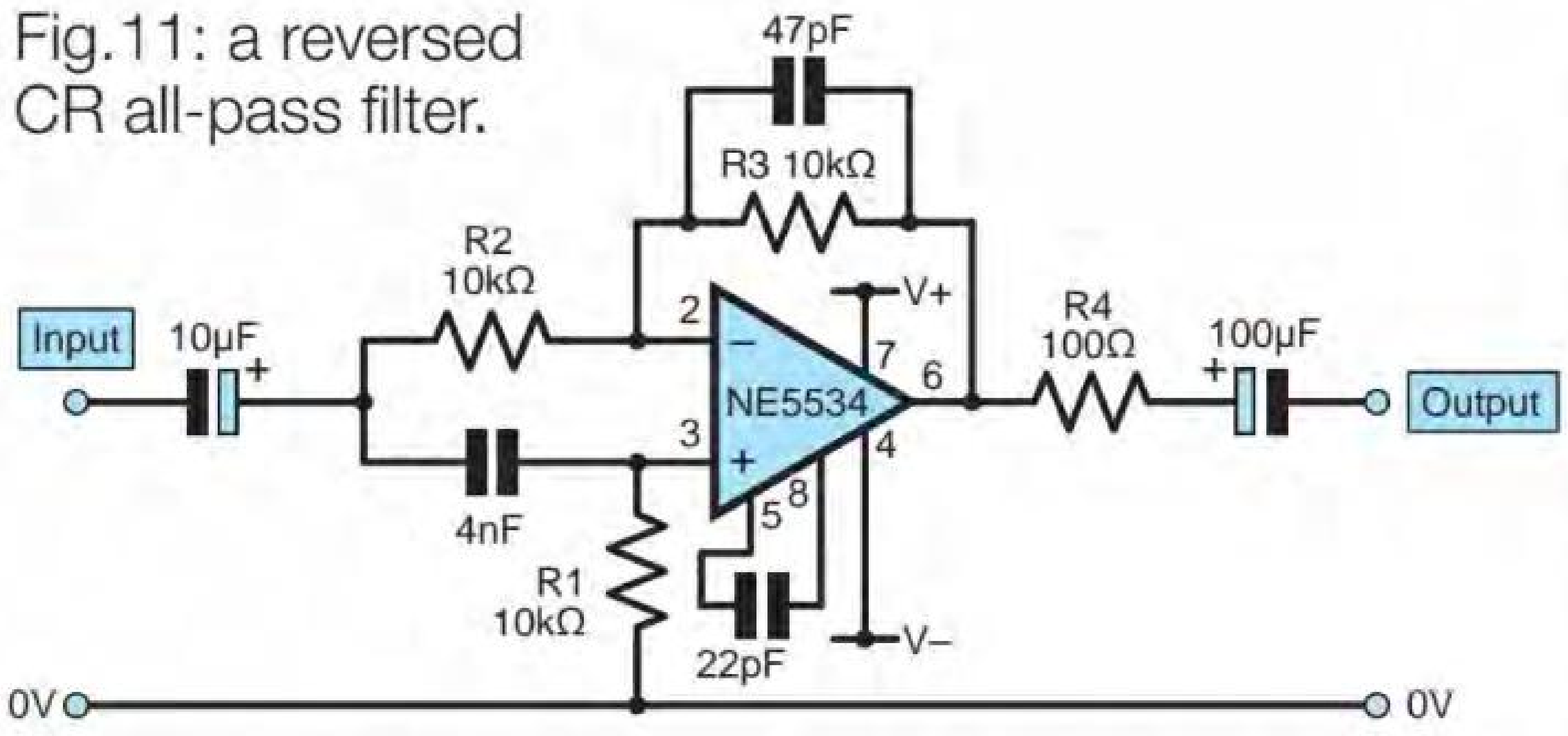
Fig.12: a complete phase rotation by cascading two all-pass stages.

it's not in some books. The curve, shown in Fig.10, has a shape more like a low-pass filter.

An article in the *Journal of the Audio Engineering Society*, December 1988, by Dene Jensen (who is an expert on group delay in audio transformers) shows what I would expect to see: a peak in group delay at the point of maximum phase shift. I expect ambiguity with audio gurus, but not in maths.

The delay introduced by an all-pass filter is not constant with frequency. Measuring it using a square wave can be misleading, because the harmonics that make up the square wave are shifted by different amounts, distorting the waveform (see Photos 3 & 4). To reveal the frequency-dependent delay (group delay), the measurement must compare the timing of different frequency components, for example using filtered or band-limited signals.

Fig.11: a reversed CR all-pass filter.


47pF
R3 10kΩ
R2
10kΩ
Input
10μF
4nF
R1
10kΩ
2
3
NE5534
7
6
4
5
8
V+
V–
22pF
R4
100Ω
100μF
Output
0V
0V

Finally, the time delay for a specific frequency and given R and C is $t = 2RC \div (1 + [2\pi fRC]^2)$. For example, Barry Porter's Active-8 loudspeaker in *ETI*, October 1984 crosses over at 3.1kHz (a typical two-way loudspeaker crossover frequency) with $R = 10k\Omega$ and $C = 1.5nF$, giving $t = 27.6\mu s$.

The position of the capacitor and resistor can be reversed if desired (see Fig.11) and this will make the phase change go the other way, which may be useful to avoid an inverting stage. The high-frequency distortion is worse with this arrangement due to the increased common-mode voltage across the op amp's input terminals.

The values of resistors R2 and R3 should be made as low as possible to minimise Johnson noise. I use 1.3kΩ, the lowest allowable with NE5532s. It is assumed that the circuit is driven from a low-impedance source, such as a preceding buffer.

Multiple-order all-pass filters with greater phase shift and time delay can be made by cascading the simple first-order circuit. If you put two in series, it goes through 180° at the main inflection point and finishes up at 360°, a complete rotation, as shown in Fig.12.

In Fig.13, four stages are cascaded for a tweeter delay; this seems to be the most popular approach. This gives a 720° phase shift and 4 × 27.6μs delays for a total of 110μs. That is sufficient to compensate for a 38mm path difference. These differences often come about either because the radiating part of one speaker driver is more recessed than another due to the shape of the speaker cones, or from driver mounting practicalities like flanges and such (often both).

Second-order all-pass filters can be made around a single op amp, but the increased noise and distortion is not worth it. There is another design we'll look at next month that uses two op amps, giving no cost saving, but the Q can be higher and the time delay frequency range is more consistent than two cascaded single stages.

High-Q phase changes can sound quite 'ringy'; could this be a useful studio effect?

There is also a discrete version, shown in Fig.14, which was used in the B&W Active 1 loudspeaker, giving reduced noise.

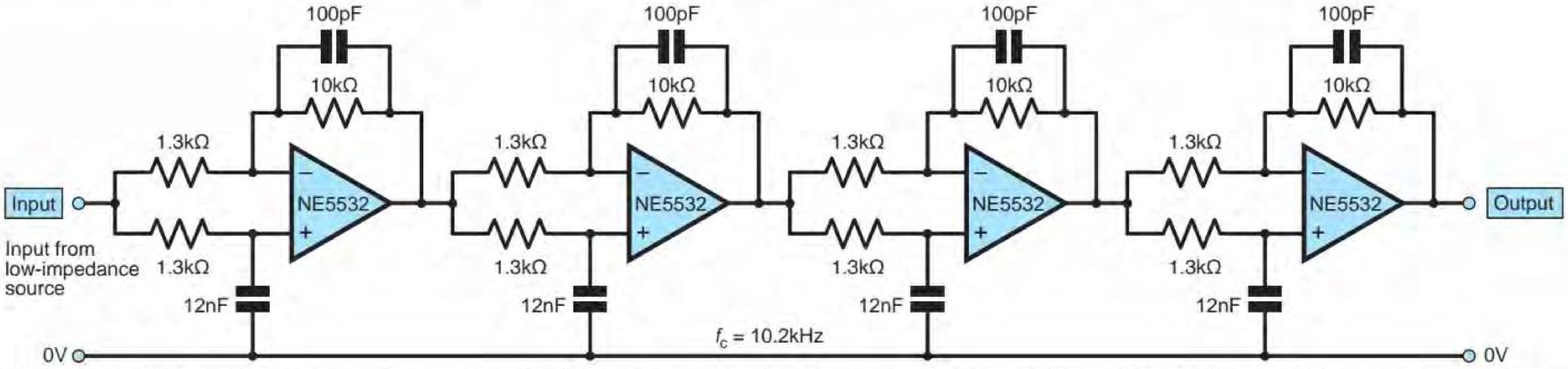

100pF
10kΩ
1.3kΩ
Input
Input from low-impedance source
1.3kΩ
NE5532
12nF
Output
f_c = 10.2kHz
0V
0V

Fig.13: all-pass filter stages can be cascaded for a longer delay in applications like driver alignment compensation.

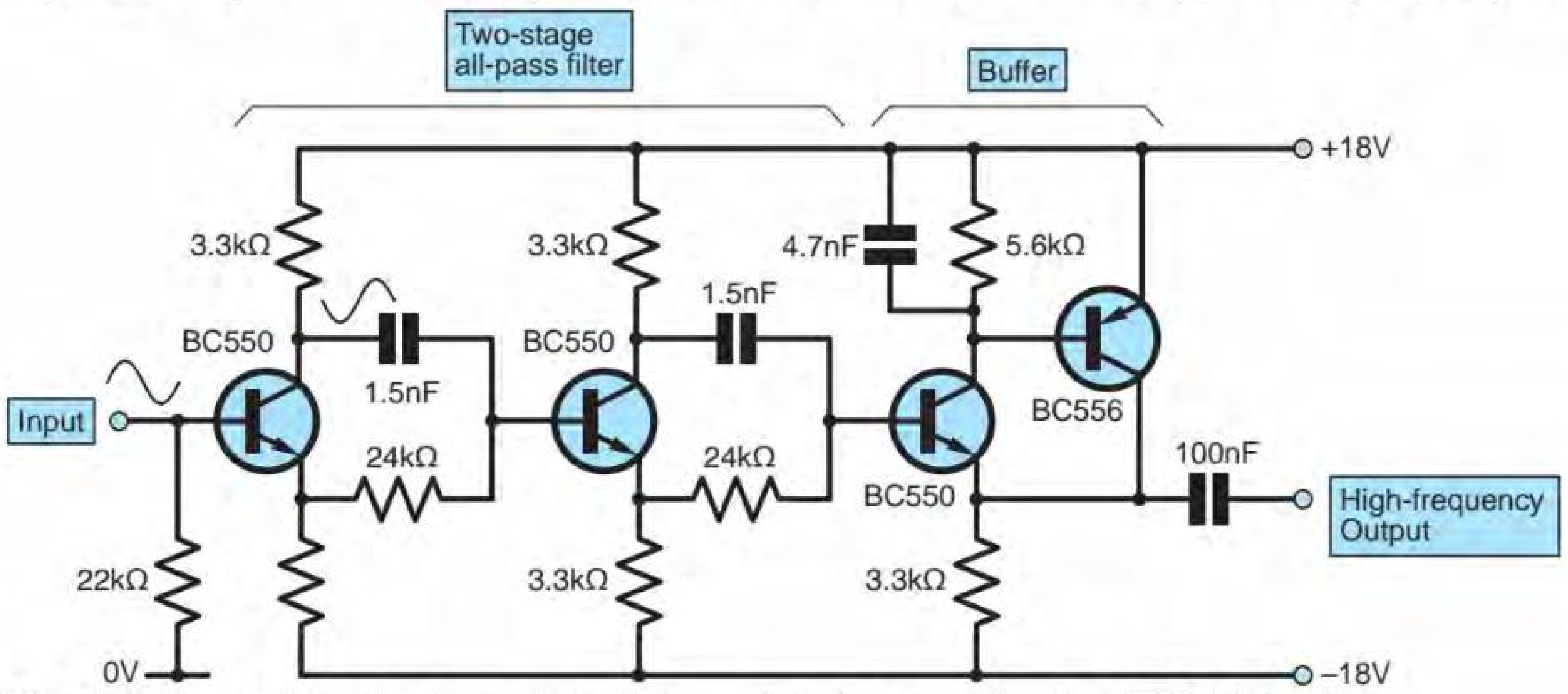

Two-stage all-pass filter
Buffer
+18V
3.3kΩ
3.3kΩ
4.7nF
5.6kΩ
1.5nF
BC550
BC550
1.5nF
Input
24kΩ
24kΩ
BC556
BC550
100nF
High-frequency Output
22kΩ
3.3kΩ
3.3kΩ
0V
–18V

Fig.14: the discrete tweeter delay circuit used in the B&W Active 1.

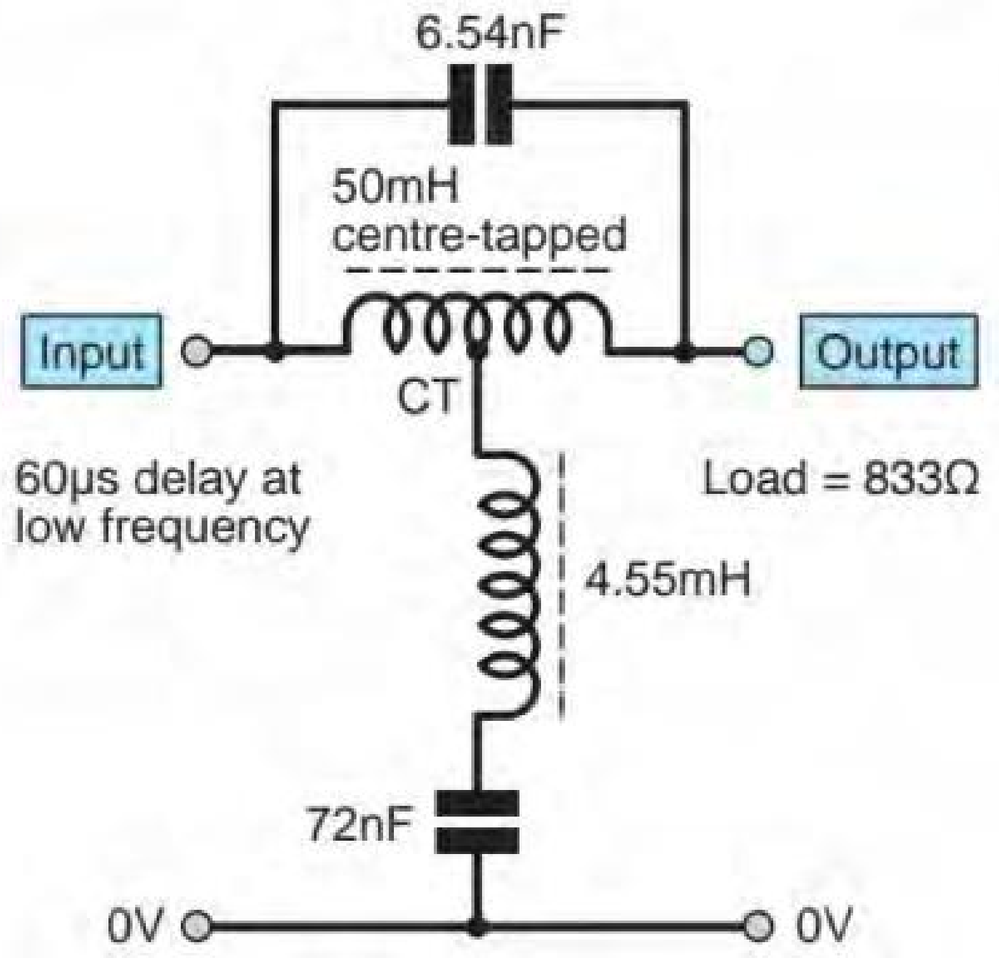

6.54nF
50mH centre-tapped
Input
Output
CT
60μs delay at low frequency
Load = 833Ω
4.55mH
72nF
0V
0V

Fig.15: a tape group delay equaliser for the Revox A700.

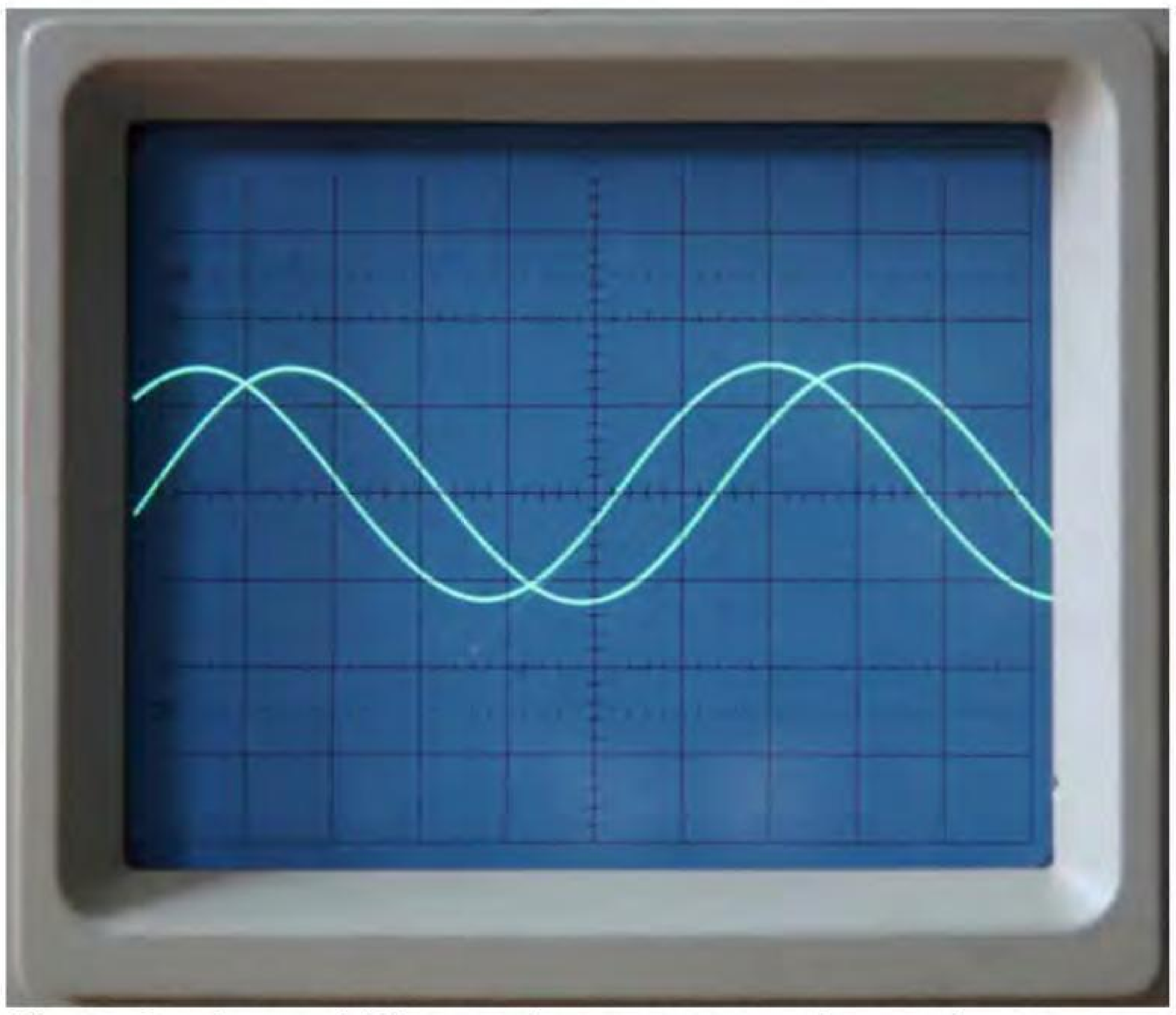


Photo 3: phase shifting a sinewave – no shape change.

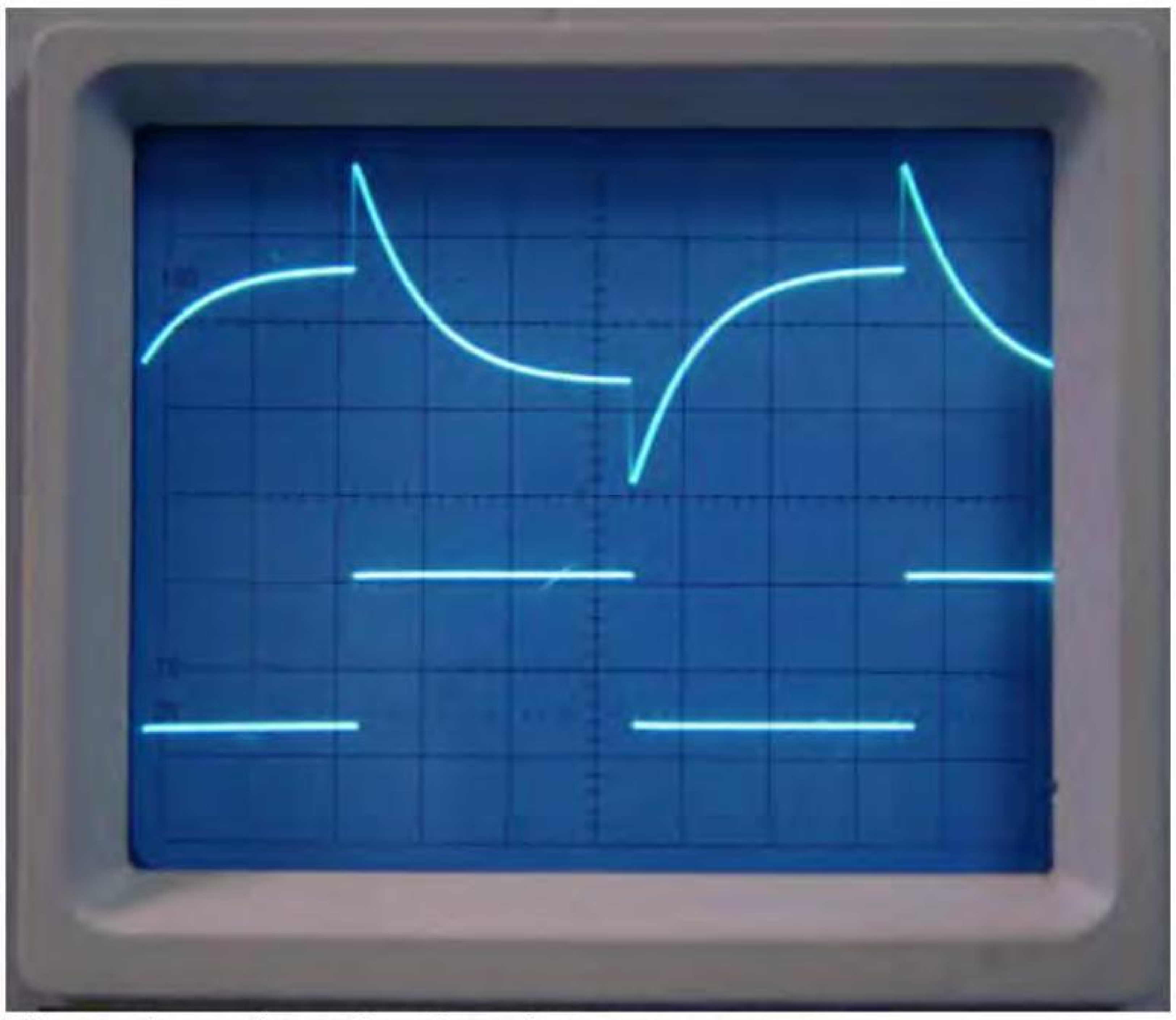
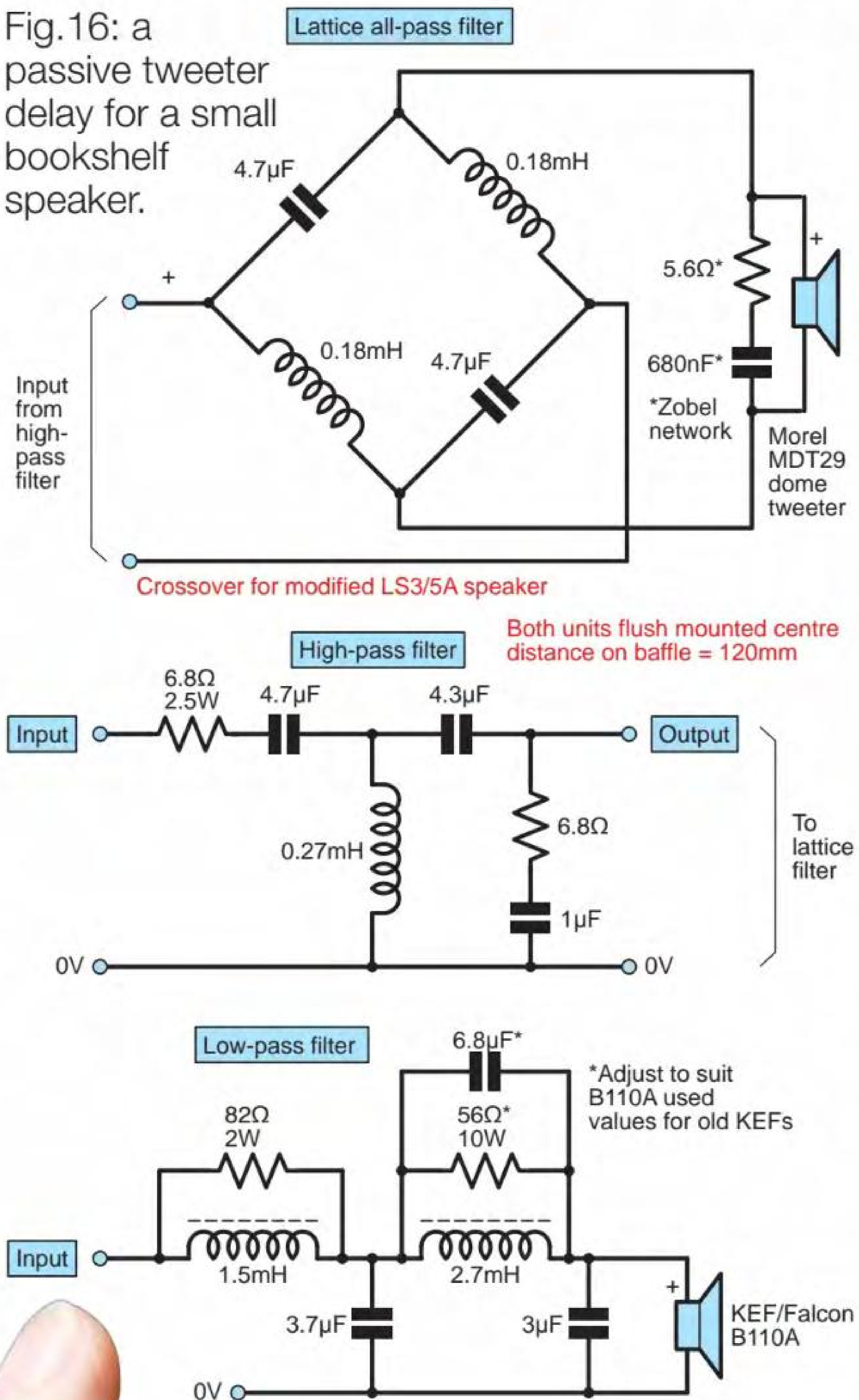


Photo 4: on the other hand, a square wave...

Fig.16: a passive tweeter delay for a small bookshelf speaker.


Lattice all-pass filter
4.7µF
0.18mH
0.18mH
4.7µF
+
Input from high-pass filter
5.6Ω*
680nF*
*Zobel network
+
Morel MDT29 dome tweeter
Crossover for modified LS3/5A speaker
High-pass filter
Both units flush mounted centre distance on baffle = 120mm
Input
6.8Ω 2.5W
4.7µF
4.3µF
Output
0.27mH
6.8Ω
1µF
To lattice filter
0V
0V
Low-pass filter
6.8µF*
*Adjust to suit B110A used values for old KEFs
82Ω 2W
56Ω* 10W
Input
1.5mH
2.7mH
3.7µF
3µF
+
KEF/Falcon B110A
0V

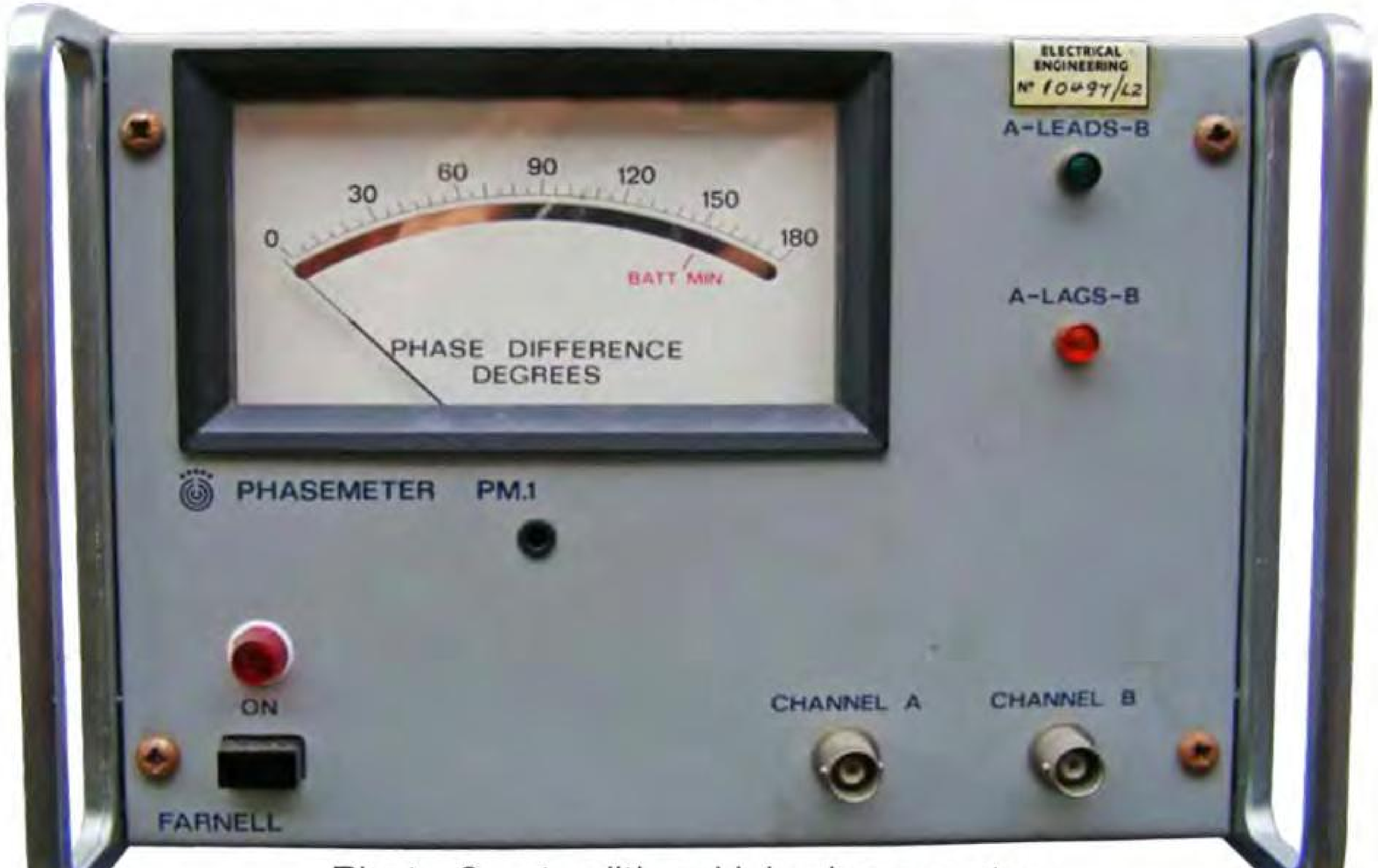

0
30
60
90
120
150
180
BATT MIN
PHASE DIFFERENCE DEGREES
ELECTRICAL ENGINEERING
Nº 10497/62
A-LEADS-B
A-LAGS-B
PHASEMETER PM.1
ON
CHANNEL A
CHANNEL B
FARNELL

Photo 6: a traditional lab phase meter.

Passive all-pass filters

In January 1981, a design to straighten out the group delay of a Revox tape recorder was published in *Hi-Fi News and Record Review*. This used a "T" network all-pass filter and is shown in Fig.15. It employs complex pot-core inductors and odd-value capacitors. Ampex ATR 100 multitrack machines had phase correction built in.

The lattice filter is another interesting topology used for passive loudspeakers and old-fashioned studio phase processors. It is illustrated in Fig.16.

Measuring phase

Photo 3 shows phase shifts on an oscilloscope, first with sinewaves and then (Photo 4) on square waves. You can see how the shape of the sinewave is preserved but the square wave's shape is definitely not!

An important and useful tool in audio mixing is the phase correlation meter shown in Photo 5, which compares the phase difference between the left and right channels independent of amplitude. When it is in the middle, that is normal stereo; when fully deflected, it is mono; and when deflected backwards, it indicates a 180° phase shift or reversed polarity.

These meters were very useful for stereo tape head azimuth alignment using a mono test tape. Photo 6 shows a traditional phase meter used in an electronics lab, which is not so useful for audio work, since it does not have a centre zero.

Lissajous figures are a classic way to show phase relationships by applying the two signals to be compared using the X and Y inputs of an oscilloscope. They can be an

0
–
PHASE
+
MYTEK USA

Photo 5: a studio phase correlation meter.

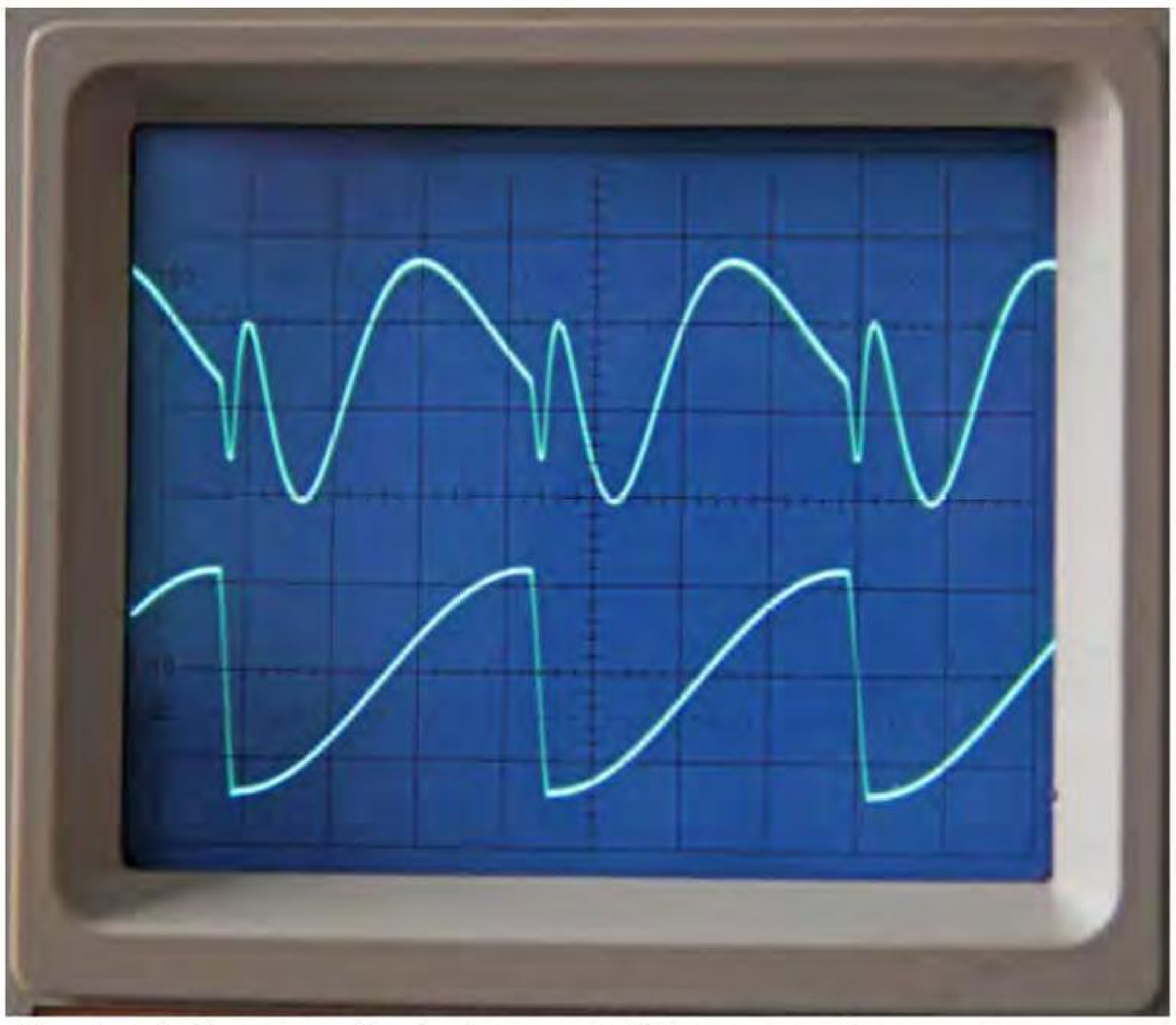


Photo 7: the result of phase-shifting a wonky ramp.

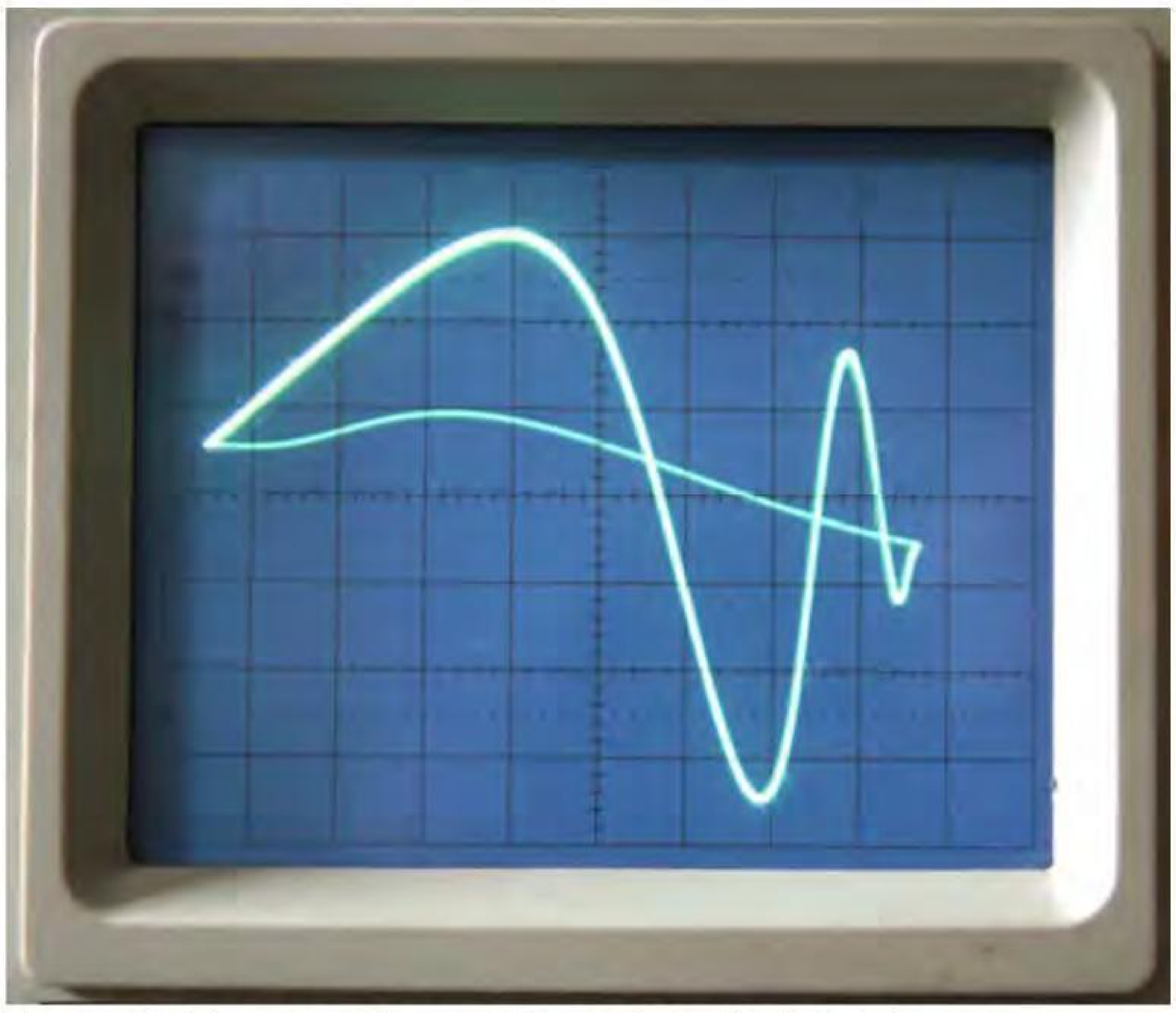


Photo 8: Lissajous figures based on that wonky ramp.

art form in their own right, as seen in Photos 7 & 8.

Phasers

The phaser effect is possibly the most common use for all-pass filters in audio, which is not surprising as it is one of the best sound effects around. Jean-Michel Jarre's groundbreaking 1976 synth hit album Oxygène was virtually built around it.

I designed and built my own dual-channel studio quality rack mount phaser, visible in Photo 9. This used a total of 12 all-pass filter stages in each channel.

Phasers produce their wonderful spacey effect by sweeping several notches up and down across the audio spectrum. The notches are created through phase cancellation. To achieve this, the direct signal is mixed with an all-pass filtered version. Two all-pass filter stages are needed for each notch, giving the required 180º for a cancellation notch with the original signal.

This element can form the basis of a comb filter. True comb filtering used for flanging has many harmonically related narrow notches; it needs a real delay line made using digital or bucket-brigade techniques (or two tape machines).

For a nice-sounding phaser, three notches are required, so six stages in total. A six-gang pot could be used to sweep the notches, but it would be an expensive component. A better solution is to use voltage control.

Voltage-controlled all-pass filters

Fig.17 shows all the filter control techniques I have found effective. The pot sections could be replaced directly by optoisolated JFETs or Vactrols (the classic LED + LDR combination), both of which are expensive, but they offer total control isolation.

A low-cost voltage-controlled resistor is the good old JFET, which most phasers use, including the Colorsound Phazex shown in Photo 10 and Fig.18. To apply these, their gate control voltage needs to be referenced to ground, so it is necessary to reverse the position of the capacitor in the filter relative to the resistor.

The JFETs have to be selected so that their pinch-off (V_p) voltages match to ensure that the filters track each other. This is a major problem with this circuit, since V_p can vary over 50% between devices. I have found the cheap Toshiba 2SK2145 SMT dual JFETs help with their better matching.

When we made phasers at Colorsound, I had to make a special JFET measuring jig (described in an earlier issue of this magazine, back when it was called *Everyday Practical Electronics*). There are two chips that can be used in a way

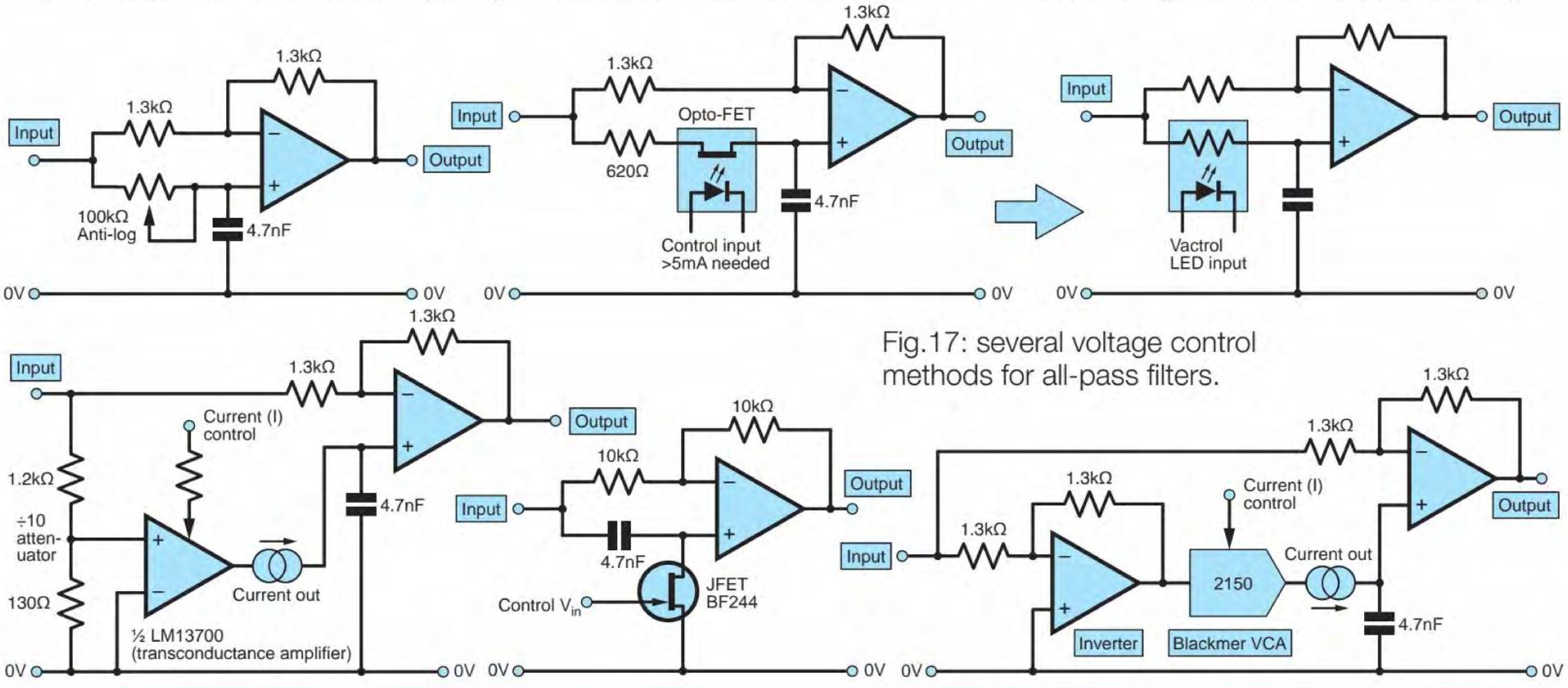

Input
1.3kΩ
1.3kΩ
Output
100kΩ
Anti-log
4.7nF
0V
0V
1.3kΩ
1.3kΩ
Input
Opto-FET
620Ω
Output
4.7nF
Control input
>5mA needed
0V
0V
Input
Output
Vactrol
LED input
0V
0V
Input
1.3kΩ
1.3kΩ
Current (I)
control
Output
1.2kΩ
÷10
atten-
uator
4.7nF
Current out
130Ω
½ LM13700
(transconductance amplifier)
0V
0V
10kΩ
10kΩ
Output
Input
4.7nF
JFET
BF244
Control V_{in}
0V
0V
1.3kΩ
1.3kΩ
1.3kΩ
Current (I)
control
Output
1.3kΩ
Input
2150
Current out
4.7nF
Inverter
Blackmer VCA
0V
0V

Fig.17: several voltage control methods for all-pass filters.

that avoids both the common mode distortion and matching problems, are the LM13700 transconductance amps, which are noisy, and Blackmer VCAs.

These devices from That Corporation avoid the noise of transconductance amps, give very low distortion and are ideal for a studio-quality phaser. However, they have an exponential control characteristic, are inverting and are very expensive.

Another chip type I'm interested in trying is the Dallas digitally controlled pot, but I suspect using those in the circuit would result in a 'zipper' noise.

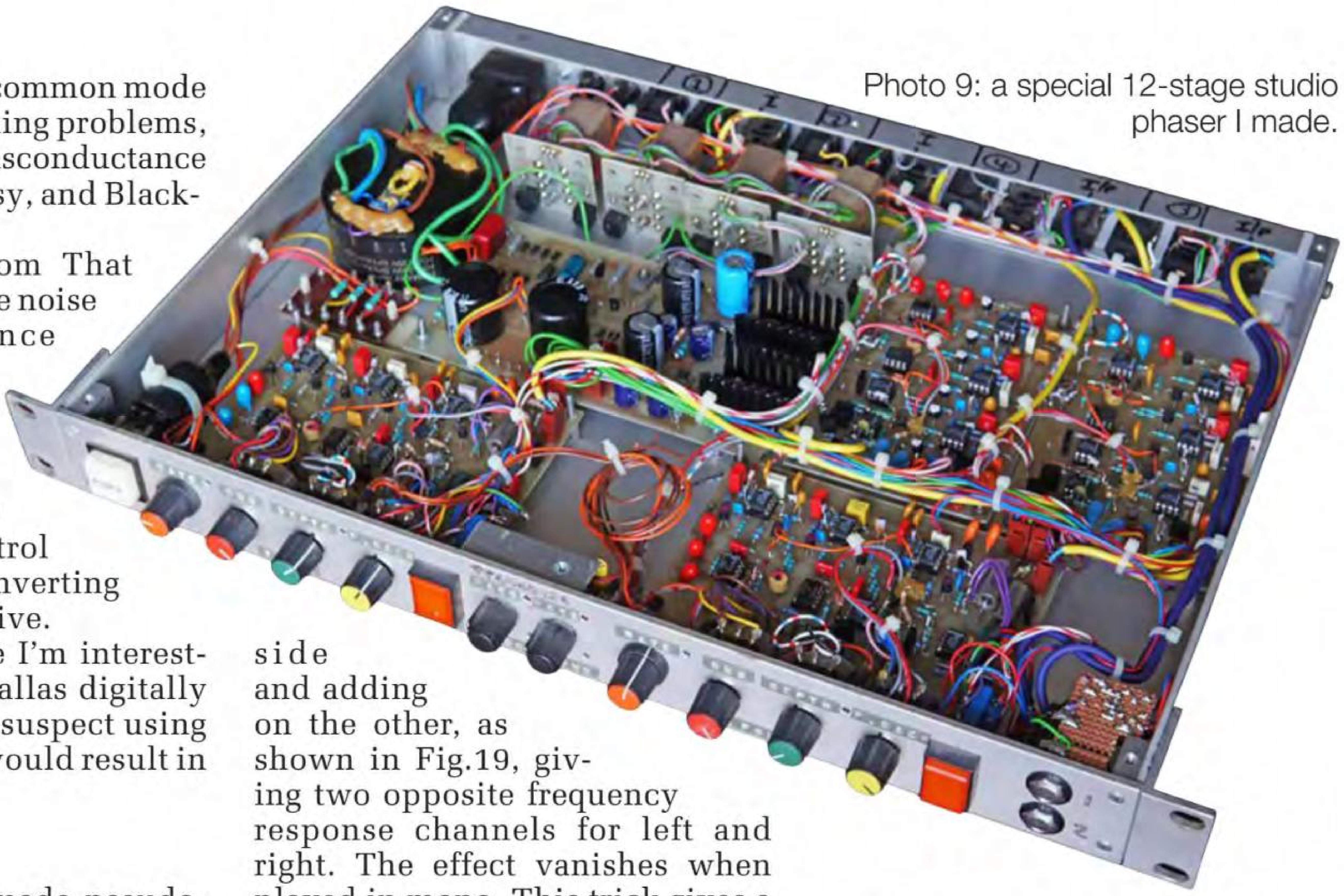


Photo 9: a special 12-stage studio phaser I made.

Stereoising

A phaser can be made pseudo-stereo by subtracting the all-passed signal from the direct signal on one side and adding on the other, as shown in Fig.19, giving two opposite frequency response channels for left and right. The effect vanishes when played in mono. This trick gives a useful 'stereoising' effect on instruments such as pianos and it does not necessarily need to be swept.

Photo 10: the Colorsound Phazex.

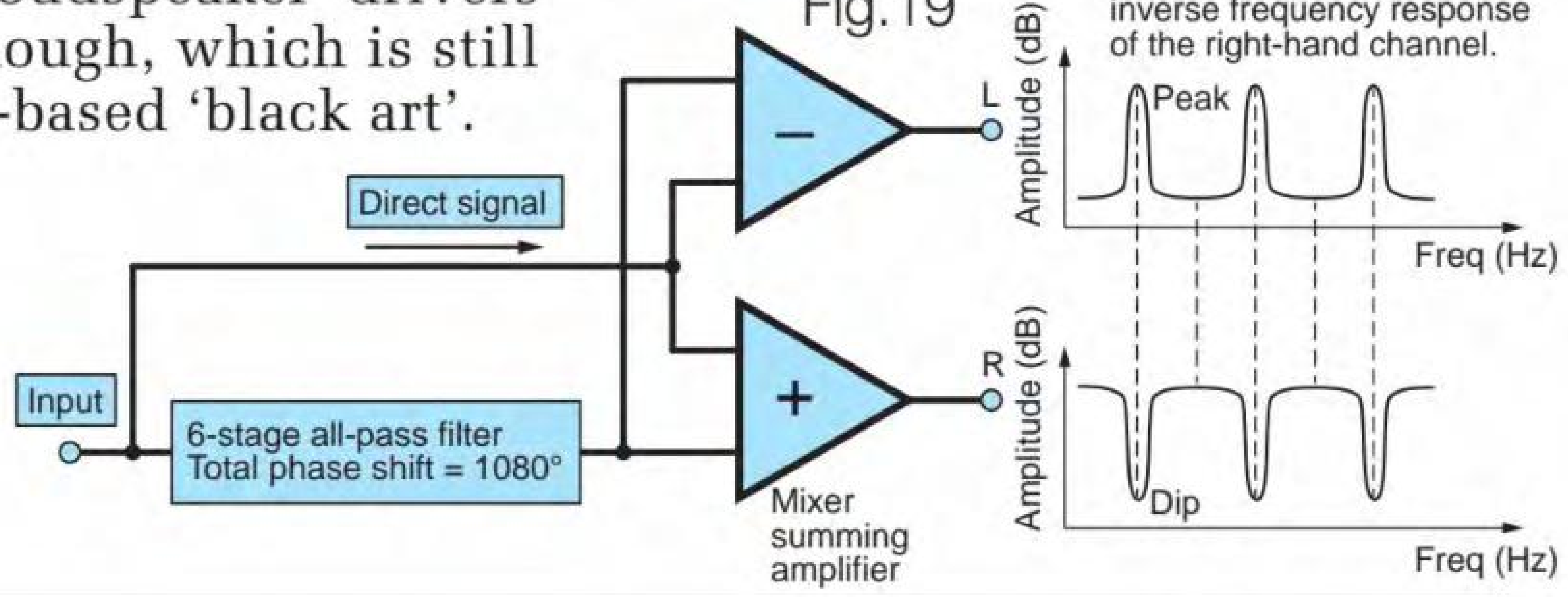
Next month

Possibly the best information available on all-pass filters is in Douglas Self's book, *The Design of Active Crossovers* (2nd Edition, 2018, ISBN 978 1 138 73303 9), which does a brilliant job of covering the electronic aspects. It does not cover combining the acoustic response of loudspeaker drivers with filters, though, which is still an experience-based 'black art'.

In part two, we'll look at loudspeakers and make an all-pass filter board with up to eight sections for multiple applications, such as a waveform 'symmetriciser', a multi-channel microphone phase adjuster and a tweeter time alignment tool, as well as some traditional phasers.

As with capacitors, there seems to be much misleading information about phase in audio out there. Watch out for being made an April fool; I found an article on designing an "all-stop" filter by Ben Sullivan in *Practical Wireless*, April 1989. **PE**

Fig.19


Left-hand channel has the inverse frequency response of the right-hand channel.
Direct signal
Input
6-stage all-pass filter
Total phase shift = 1080°
–
+
L
R
Mixer summing amplifier
Amplitude (dB)
Peak
Freq (Hz)
Amplitude (dB)
Dip
Freq (Hz)

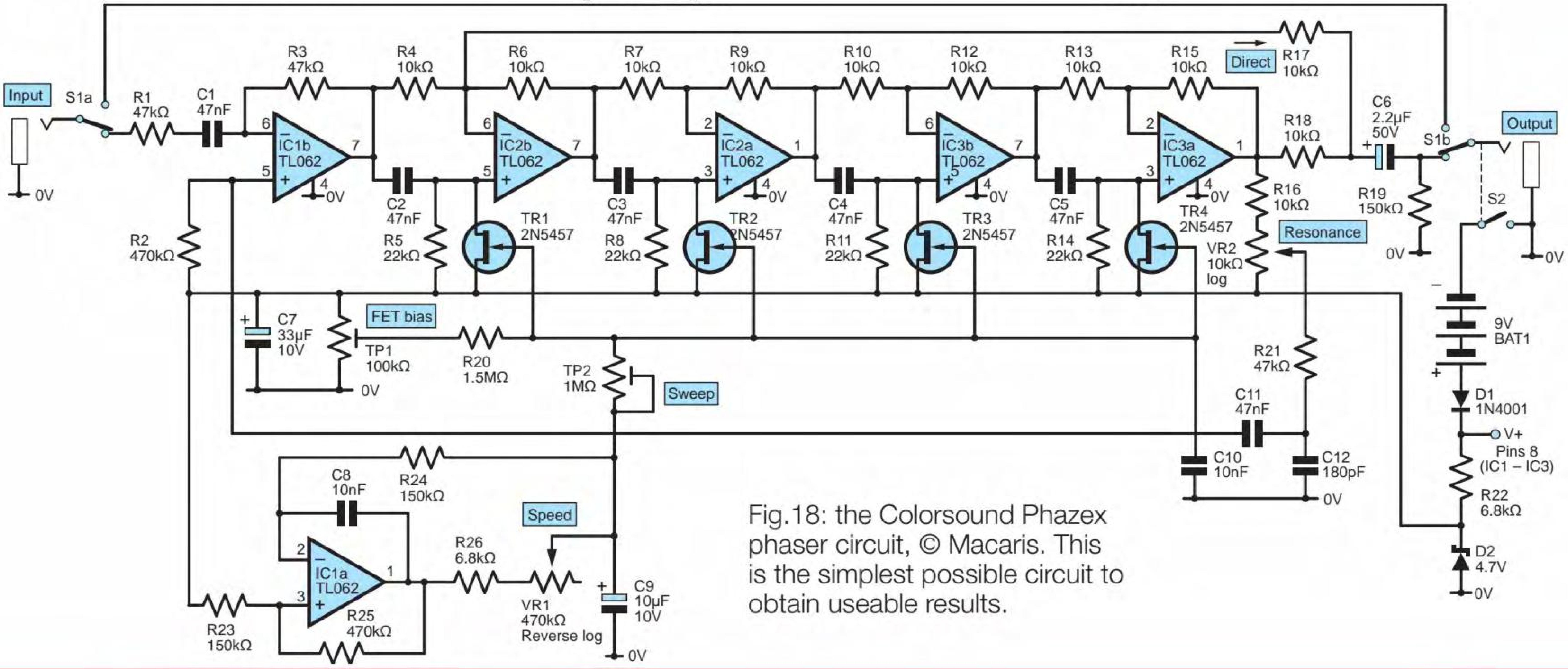

Input
S1a
R1 47kΩ
C1 47nF
0V
R2 470kΩ
R3 47kΩ
IC1b TL062
R4 10kΩ
C2 47nF
R5 22kΩ
IC2b TL062
TR1 2N5457
R6 10kΩ
R7 10kΩ
C3 47nF
R8 22kΩ
IC2a TL062
TR2 2N5457
R9 10kΩ
R10 10kΩ
C4 47nF
R11 22kΩ
IC3b TL062
TR3 2N5457
R12 10kΩ
R13 10kΩ
C5 47nF
R14 22kΩ
IC3a TL062
TR4 2N5457
R15 10kΩ
Direct
R17 10kΩ
R18 10kΩ
C6 2.2µF 50V
S1b
Output
R16 10kΩ
R19 150kΩ
Resonance
VR2 10kΩ log
S2
C7 33µF 10V
FET bias
TP1 100kΩ
R20 1.5MΩ
TP2 1MΩ
Sweep
R21 47kΩ
C11 47nF
C10 10nF
C12 180pF
9V BAT1
D1 1N4001
V+
Pins 8 (IC1 – IC3)
R22 6.8kΩ
D2 4.7V
C8 10nF
R24 150kΩ
IC1a TL062
R23 150kΩ
R25 470kΩ
R26 6.8kΩ
Speed
VR1 470kΩ Reverse log
C9 10µF 10V
0V

Fig.18: the Colorsound Phazex phaser circuit, © Macaris. This is the simplest possible circuit to obtain useable results.

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
APRIL 2026		
Versatile Battery Checker PCB	11104251	£3.99
Versatile Battery Checker front panel (black, 0.8mm)	11104252	£4.99
Power Rail Probe PCB	P9058-1-C	£3.99
Pico 2 Audio Analyser PCB/panel (black)	04107231	£3.99
MARCH 2026		
USB-Programmable Frequency Divider PCB (blue)	04108241	£3.99
Rotating Light PCB (black)	09101251	£2.99
Arduino Uno Game Cartridge PCB (Cool Beans)	CBmar26uno	£3.99
Arduino Nano Game Cartridge PCB (Cool Beans)	CBmar26nano	£2.99
Silent Stomp Switch PCB pair (Audio Out)	AO-MAR26-*	£3.99
FEBRUARY 2026		
Power LCR Meter PCB	04103251	£7.99
Versatile Waveform Generator PCB	04104251	£5.99
JANUARY 2026		
High-Bandwidth Differential Probe	9015-D	£5.99
NFC-Programmable IR Remote Keyfob	15109231	£2.99
Pre-driver Overdrive Pedal main PCB	AO-JAN26-1	£4.99
Updated jack PCB for the Pre-driver	AO-JAN26-2	£3.99
DECEMBER 2025		
Variable Speed Drive (black)	11111241	£11.99
Digital Capacitance Meter	04111241	£5.99
Battery-Powered Model Train transmitter	09110241	£3.99
Battery-Powered Model Train through-hole receiver	09110242	£3.99
Battery-Powered Model Train SMD receiver	09110243	£3.99
Battery-Powered Model Train battery charger	09110244	£3.99
Transmanium Fuzz Box jack PCB	AO-DEC-25	£3.99
NOVEMBER 2025		
Capacitor Discharger	9047-01	£4.99
5MHz, 40A Current Probe	9049-01	£5.99
Transmanium Fuzz Box main PCB	AO-NOV-25	£4.99
OCTOBER 2025		
Seaside Sound Simulator PCB (blue)	01111241	£7.99
Fender Jazz Bass pickguard	23109241	£5.99
J&D Luthiers T-Style pickguard	23109242	£5.99
Music Man Stingray pickguard	23109243	£5.99
Fender Telecaster pickguard	23109244	£4.99
3D Printer Filament Dryer control PCB	28110241	£4.99
SEPTEMBER 2025		
Automatic LQ Meter PCB	CSE240203A	£5.99
Automatic LQ Meter front panel (black)	CSE240204A	£5.99
Compact Hi-Fi Headphone Amplifier PCB (blue)	01103241	£4.99
Discrete Ideal Bridge Rectifier TH PCB	18108241	£2.99
Discrete Ideal Bridge Rectifier SMD PCB	18108242	£2.99
Dual Rail Supply Protector PCB	18109241	£3.99
Audio Out Cascode Amplifier PCB	CASCODEA	£4.99
Audio Out Cascode Amplifier Power Supply PCB	CASCODEP	£4.99
AUGUST 2025		
Micromite Explore-40 PCB	07106241	£2.99
Micromite Explore-40 Audio Breakout PCB	07101222	£2.99
8-Channel Learning IR Remote Receiver PCB	15108241	£5.99
FlexiDice PCB set	08107241-2	£4.99
JULY 2025		
Compact OLED Clock & Timer PCB	19101231	£4.99
180-230V DC Motor Speed Controller PCB	11104241	£9.99
Mains Power-Up Sequencer PCB (same as Feb '25)	10108231	£11.99

PROJECT	CODE	PRICE
JUNE 2025		
Styloclone (case version)	23106241	£6.99
Styloclone (standalone version)	23106242	£7.99
Dual Mini LED Dice (TH LEDs, white solder mask)	08103241W	£2.99
Dual Mini LED Dice (TH LEDs, black solder mask)	08103241B	£2.99
Dual Mini LED Dice (SMD LEDs, white solder mask)	08103242W	£2.99
Dual Mini LED Dice (SMD LEDs, black solder mask)	08103242B	£2.99
Dual Hybrid Supply regulator board	18107211	£5.99
Dual Hybrid Supply front panel control board	18107212	£2.99
Dual Hybrid Supply CPU board	01106193	£3.99
Dual Hybrid Supply LCD adaptor	01106196	£2.99
MAY 2025		
Skill Tester 9000	08101241	£9.99
USB-C Serial Adaptor (black solder mask)	24106241	£2.99
DC Supply Protector (SMD adjustable version)	08106241	£3.99
DC Supply Protector (TH adjustable version)	08106242	£3.99
DC Supply Protector (TH fixed version)	08106243	£3.99
APRIL 2025		
ESR Test Tweezers PCB set	04105241+	£5.99
Fan Speed Controller Mk2	10104241	£4.99
Calibrated Microphone (SMD version)	01108231	£3.99
Calibrated Microphone (TH version)	01108232	£3.99
MARCH 2025		
Pico Gamer	08104241	£9.99
Compact Frequency Divider	04112231	£4.99
FEBRUARY 2025		
Laser Communicator transmitter	16102241	£3.99
Laser Communicator receiver	16102242	£3.99
Model Railway Points Controller transmitter	09101241	£3.99
Model Railway Points Controller receiver	09101242	£2.99
Mains Power-Up Sequencer	10108231	£11.99
JANUARY 2025		
USB to PS/2 Converter for VGA PicoMite	07111231	£3.99
ps2x2pico USB to PS/2 Converter	07111232	£3.99
Compact Microphone Preamplifier (case version)	01110231	£5.99
Compact Microphone Preamplifier (integrated version)	01110232	£5.99
DECEMBER 2024		
Raspberry Pi Clock Radio PCB pair	19101241/2	£12.99
VGA PicoMite (blue)	07107221	£5.99
Secure Remote Switch receiver	10109231	£4.99
Secure Remote Switch transmitter (module version)	10109232	£2.99
Secure Remote Switch transmitter (discrete version)	10109233	£2.99
NOVEMBER 2024		
Ideal Diode Bridge Rectifiers (28mm square)	18101241	£3.99
Ideal Diode Bridge Rectifiers (22mm square)	18101242	£3.99
Ideal Diode Bridge Rectifiers (4-pin SIL)	18101243	£3.99
Ideal Diode Bridge Rectifiers (tiny SOT-23)	18101244	£3.99
Ideal Diode Bridge Rectifiers (standalone SMD)	18101245	£4.99
Ideal Diode Bridge Rectifiers (standalone TH; 2oz copper)	18101246	£6.99
Multi-channel Volume Control (volume PCB)	01111221	£4.99
Multi-channel Volume Control (control PCB)	01111222	£4.99
Multi-channel Volume Control (OLED PCB, black)	01111223	£4.99
Coin Cell Emulator (black)	18101231	£3.99
OCTOBER 2024		
TQFP-32 Programming Adaptor	24108231	£4.99
TQFP-44 Programming Adaptor	24108232	£4.99
TQFP-48 Programming Adaptor	24108233	£4.99
TQFP-64 Programming Adaptor	24108234	£4.99
2m VHF FM Test Signal Generator	06107231	£4.99

PCBs for most recent *PE/EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Magazine back issues are available – see https://www.electronpublishing.com/product-category/pe-back-issues-print-and-digital/

Double-sided | plated-through holes | solder mask

PROJECT	CODE	PRICE
SEPTEMBER 2024		
PIC Programming Adaptor	24105231	£4.99
Salad Bowl Speaker crossover	01109231	£7.99
AUGUST 2024		
Arduino ESR Meter combined L/C-ESR	04106182	£9.99
Arduino ESR Meter separate ESR	04106183	£7.99
Watering System Controller	15110231	£11.99
JULY 2024		
Dynamic NFC Tag (22 × 31mm)	06101231	£5.95
Dynamic NFC Tag (86 × 54mm)	06101232	£7.95
Reciprocal Frequency Counter	CSE230101C	£8.95
Reciprocal Frequency Counter front panel (black mask)	CSE230102	£7.95
Pi Pico-based Thermal Camera	04105231	£6.95
JUNE 2024		
Loudspeaker Test Jig	04106231	£12.95
MAY 2024		
GPS-Disciplined Oscillator	04103231	£12.95
Dual RF Amplifier	CSE220602A	£5.95
Songbird	08103231	£7.95
APRIL 2024		
Basic RF Signal Generator	CSE221001	£8.99
Basic RF Signal Generator front panel (black solder mask)	CSE220902B	£9.99
Silicon Chirp Cricket	08101231	£7.99
Test Bench 'Swiss Army Knife'	04110221	£9.99
MARCH 2024		
Digital Volume Control Potentiometer SMD version	01101231	£8.95
Digital Volume Control Potentiometer through-hole version	01101232	£9.95
Advanced SMD Test Tweezers body	04106221	£6.95
Advanced SMD Test Tweezers legs (pair)	04106212	£7.95
FEBRUARY 2024		
Active Mains Soft Starter	10110221	£10.95
JANUARY 2024		
Q Meter	CSE220701	£8.95
Q Meter front panel (black solder mask)	CSE220704	£7.95
Raspberry Pi Pico W BackPack	07101221	£8.95
DECEMBER 2023		
Digital Boost Regulator	24110224	£8.95
Dual-Channel Power Supply for Breadboards	04112221	£11.95
Display Adaptor for the Breadboard PSU	04112222	£9.95
NOVEMBER 2023		
LC Meter Mk3	CSE220503C	£9.95
DC Supply Filter for vehicles	08108221	£8.95
Discrete Audio Op Amp PCB	AO1-JUL23	£9.95
Discrete Audio Op Amp PCB + essential components	AO2-JUL23	£17.95
OCTOBER 2023		
Buck/Boost Charger Adaptor	14108221	£9.95
PIC Breakout Board for SOIC parts	24110225	£5.95
PIC Breakout Board for DIP parts	24110222	£5.95
AVR64DD32 Breakout board	24110223	£5.95
Automatic Train Controller	09109221	£6.95
Chuff Sound module	09109222	£6.95
SEPTEMBER 2023		
Mini LED Driver	16106221	£7.95
New GPS-Synchronised Clock	19109221	£9.95

PROJECT	CODE	PRICE
AUGUST 2023		
110dB RF Attenuator	CSE211003	£9.95
Universal Battery Charge Controller (2023 update)	14107192	£9.95
Wide-Range OhmMeter	04109221	£12.95
JULY 2023		
Multimeter Checker/Calibrator	04107221	£10.95
JUNE 2023		
Arduino Programmable Load	04105221	£8.95
Buck-Boost LED Driver	16103221	£8.95
MAY 2023		
Precision AM-FM DDS Signal Generator	CSE211002	£14.95
Improved SMD Test Tweezers programmed PIC	0410621PIC	£12.95
AO 2x-dual-to-quad through-hole (pack of five)	AO1-MAY23	£6.95
AO 2x-dual-to-quad SMD (pack of five)	AO2-MAY23	£6.95
AO 2x-single-to-dual through-hole (pack of five)	AO3-MAY23	£6.95
AO 2x-single-to-dual SMD (pack of five)	AO4-MAY23	£6.95
APRIL 2023		
500W Amplifier Module	see p.22, April 2023	
Clipping Indicator (per channel)	01112211	£7.95
CD Welder Power Supply (one needed)	29103221	£9.95
CD Welder Controller (one needed)	29103222	£9.95
CD Welder Energy Storage module (several needed)	29103223	£7.95
AO Universal Dual Op Amp Board	AO1-APR23	£9.95
AO Stereo RIAA precision passives kit for dual op amp	AO2-APR23	£8.95
MARCH 2023		
Pico BackPack	07101221	£9.95
Semaphore Signal (controller)	09103221	£7.95
Semaphore Signal (blade)	09103222	£5.95
FEBRUARY 2023		
Tesla Coil driver board	26102221	£9.95
Tesla Coil potentiometer board	26102222	£5.95
Cooling Fan Controller & Loudspeaker Protector	01102221	£8.95
Remote Gate Controller	11009121	£12.95
JANUARY 2023		
Classic LED Metronome – 8-LED	23111211	£7.95
Classic LED Metronome – 10-LED	23111212	£8.95
Multi-Channel Speaker Protector – 6 channel	01101221	£9.95
Multi-Channel Speaker Protector – 4 channel	01101222	£7.95
Remote Control Range Extender – IR-to-UHF	15109212	£5.95
Remote Control Range Extender – UHF-to-IR	15109211	£7.95
AO Universal Single Op Amp Board	AO1-JAN23	£7.95
DECEMBER 2022		
Hummingbird Amplifier	01111211	£9.95
SMD Trainer PCB	29106211A	£8.95
NOVEMBER 2022		
USB Cable tester – main PCB	04108211	£12.95
USB Cable tester – front panel	04108212	£5.95
USB Cable tester – optional panel	SC5970	£5.95
Model Railway Carriage Lights – PCB	09109211	£6.95
AO trnsfmr PCB – standard VTX-A range	VTX-101-007	£6.95
AO trnsfmr PCB – dual-outline VTX102-3001/101-3002	VTX-Dual	£6.95
OCTOBER 2022		
SMD Test Tweezers – PCB and pair of tweezer arms	04106211-2	£11.95
SMD Test Tweezers – programmed PIC12F1572-I/SN	0410621A	£7.95
Tele-com	12110211	£12.95

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.

Orders and payment should be sent to:

Practical Electronics, Electron Publishing Ltd
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel 07918 614662 Email: pe@pemag.au
Online Shop: www.electronpublishing.com

Sorry, cheques cannot be accepted as we are based in Australia and our UK bank account does not allow cheque deposits.

NOTE: Most boards are in stock and sent within seven days of order receipt. Please allow up to 28 days delivery if we need to restock.

Double-sided | plated-through holes | solder mask

PROJECT	CODE	PRICE
SEPTEMBER 2022		
Touchscreen Digital Preamp – main board	01103191	£12.95
Touchscreen Digital Preamp – adaptor board pair	01103192	£5.95
20A DC Motor Speed Controller	11006211	£9.95
AUGUST 2022		
Multi-purpose Battery Manager – I/O Expander	11104212	£5.95
Multi-purpose Battery Manager – Switch Module	11104211	£8.95
Simple MIDI Music Keyboard (for 8 switches)	23101213	£6.95
Nano Pong	08105212	£7.95
JULY 2022		
Silicon Labs AM/FM/SW Radio	CSE210301C	£10.95
Level Crossing Controller	09108211	£6.95
JUNE 2022		
Full-wave Motor Speed Controller	1010221	£8.95
PIC Programming Helper for 8-pin PICs only	24106211	£7.95
PIC Programming Helper for 8, 14 or 20-pin PICs	24106212	£10.95
Advanced GPS Computer	05102211	£9.95
MAY 2022		
Bus board PCB for Analogue Vocoder	AO1-MAY22	£10.95
Complete set of 14 PCBs for Analogue Vocoder	AO2-MAY22	£97.95
Programmed EEPROM for Digital FX Unit	FX1-MAY22	£10.95
Programmed PIC for Digital FX Unit w/ potentiometer	FX2-MAY22	£8.95
APRIL 2022		
64-key MIDI Matrix shield	23101211	£8.95
64-key MIDI Matrix switch board	23101212	£11.95
High-current Battery Balancer	14102211	£10.95
Digital FX Unit – using potentiometer	01102211	£9.95
Digital FX Unit – using BCD switch	01102212	£9.95
Universal Audio PSU	AO1-APR22	£11.95
MARCH 2022		
Mini Isolated Serial Link	24102211	£5.95
Busy Loo Indicator	16112201	£5.95
Analogue Vocoder – Band-pass filter board	AO1-MAR22	£9.95
Analogue Vocoder – HP/LP filter board	AO2-MAR22	£9.95
FEBRUARY 2022		
Arduino-based Power Supply	18106201	£9.95
Battery Monitor Logger	11106201	£10.95
Electronic Wind Chimes	23011201	£10.95
Analogue Vocoder – Driver Amplifier	AO-FEB22	£8.95
JANUARY 2022		
Vintage battery Radio Li-ion Power Supply	11111201	£9.95
MiniHeart: A Miniature Heartbeat Simulator	01109201	£8.95
DECEMBER 2021		
AM/FM/SW Digital Receiver	CSE200902A	£13.95
Balanced Input and Attenuator for USB CODEC	01106202	£11.95
NOVEMBER 2021		
Dual Battery Lifesaver	11111202	£6.95
OCTOBER 2021		
Mini Wi-Fi LCD BackPack	24106201	£8.95

PROJECT	CODE	PRICE
SEPTEMBER 2021		
USB SuperCodec PCB	01106201	£14.95
Audio DDS Oscillator PCB	01110201	£5.95
Audio DDS Oscillator rotary encoder	01110201-ENC	6.95
Programming Adaptor Board for Audio DDS Oscillator	01110202	£5.95
High-power Ultrasonic Cleaner main PCB	04105201	£14.95
High-power Ultrasonic Cleaner front-panel PCB	04105202	
Night Keeper Lighthouse PCB	08110201	£5.95
AUGUST 2021		
Ol' Timer PCB	19104201	£11.95
Ol' Timer 8x8 RGB LED module using WS2812B	19104201-88	£8.95
Ol' Timer set of acrylic case pieces and spacer	19104201-ACR	£8.75
Ol' Timer DS3231 RTC IC wide SOIC-16	19104201-RTC	£5.95
Wideband Digital RF Power Meter	04106201	£9.75
Switchmode 78xx regulators ***(PACK of 5!)***	18105201	£7.95
Cool Beans SMAD display	CB-AUG21	£11.95
JULY 2021		
ATtiny816 Breakout / Dev Board w/ Capacitive Touch	24110181	£9.75
IR Remote Control Assistant (Jaycar version)	15005201	£8.95
IR Remote Control Assistant (Altronics version)	15005202	£8.95
PIC18F Development Board	PNM-JUL21	£12.95
Microphone Preamplifier	AO-JUL21	£11.95

For the many earlier PCBs that we stock please see the PE website: www.electronpublishing.com

PE/EPE PCB SERVICE

Order Code	Project	Quantity	Price

Name ....
Address ....
....
Tel ....
Email ....
Card No ....
Valid From ....Expiry Date ....
Card Security No ....

You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com
No need to cut your issue – a copy of this form is just as good!

Silicon Chip kits & parts for projects in this issue (posted from Australia):

Versatile Battery Checker kit (includes PCB, parts, case & panel)	https://siliconchip.au/Shop/20/7465	AUD $78 (about £40.00) + P&P
Pico 2 Audio Analyser short-form kit (without case)	https://siliconchip.au/Shop/20/6772	AUD $60 (about £31.00) + P&P

20% VAT is included in the above price (it will be removed for customers outside the UK). Postage & packing cost is typically around £12-15.
Please allow 1-2 weeks for delivery (longer in some cases).

CLASSIFIED ADVERTISING

Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value.

The rate for semi-display space is £10 per centimetre high, with a minimum height of 2·5cm. All semi-display advertisements have a width of 5.5cm. The prepaid rate for classified adverts is 40p per word (minimum 12 words).

For rates, further information on display and classified advertising or to book an ad, please email us at **pe@pemag.au**

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for more than thirty years.

W5SWL Electronics

Technical Products and Small Component Parts, RF and Audio Connectors, Hobby Electronics, Ham Radio to Microwave Products, General Hobby Supplies, Technical Books, Tools, and More. **Located in the United States.**

Visit **www.W5SWL.com**

BOWOOD ELECTRONICS LTD

Suppliers of Electronic Components

www.bowood-electronics.co.uk

Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222

Send large letter stamp for Catalogue

COAST ELECTRONICS

BREAKOUTS-COMPONENTS-CONTRACT DESIGN-3D PRINTER PARTS-MUSICAL-MICROCONTROLLERS

WWW.COASTELECTRONICS.CO.UK

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS? For free stock list and/or advice, please contact me: geoffdavies337@gmail.com Telephone: 01788 574774

PIC DEVELOPMENT KITS, DTMF kits and modules, CTCSS Encoder and Decoder/Display kits. Visit www.cstech.co.uk

Electrical Industries Charity (EIC)

We help people working in the electrical, electronics and energy community as well as their family members and retirees.

We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.

www.electricalcharity.org

electrical industries charity

ADVERTISING INDEX

AOSHOP	57
ENIGMA COMPONENT SHOP	12
MICROCHIP TECHNOLOGY	Outer back cover
PCBWAY	Inner front cover
PDF BACK ISSUES	13, 53
POLABS	47
SILICON CHIP MAGAZINE	32

Advertising mailing address

Electron Publishing Australia
PO Box 194
Matraville NSW 2036
Australia

Email pe@pemag.au

Web www.electronpublishing.com

For editorial contact details, see page 2.

Next Month – in the May issue

SSB RECEIVER
14.102.000 MHz
LSB
ANT TUNE
RF GAIN
BAND
SQUELCH
VOLUME
PHONES
USB
LSB
ON
SLOW

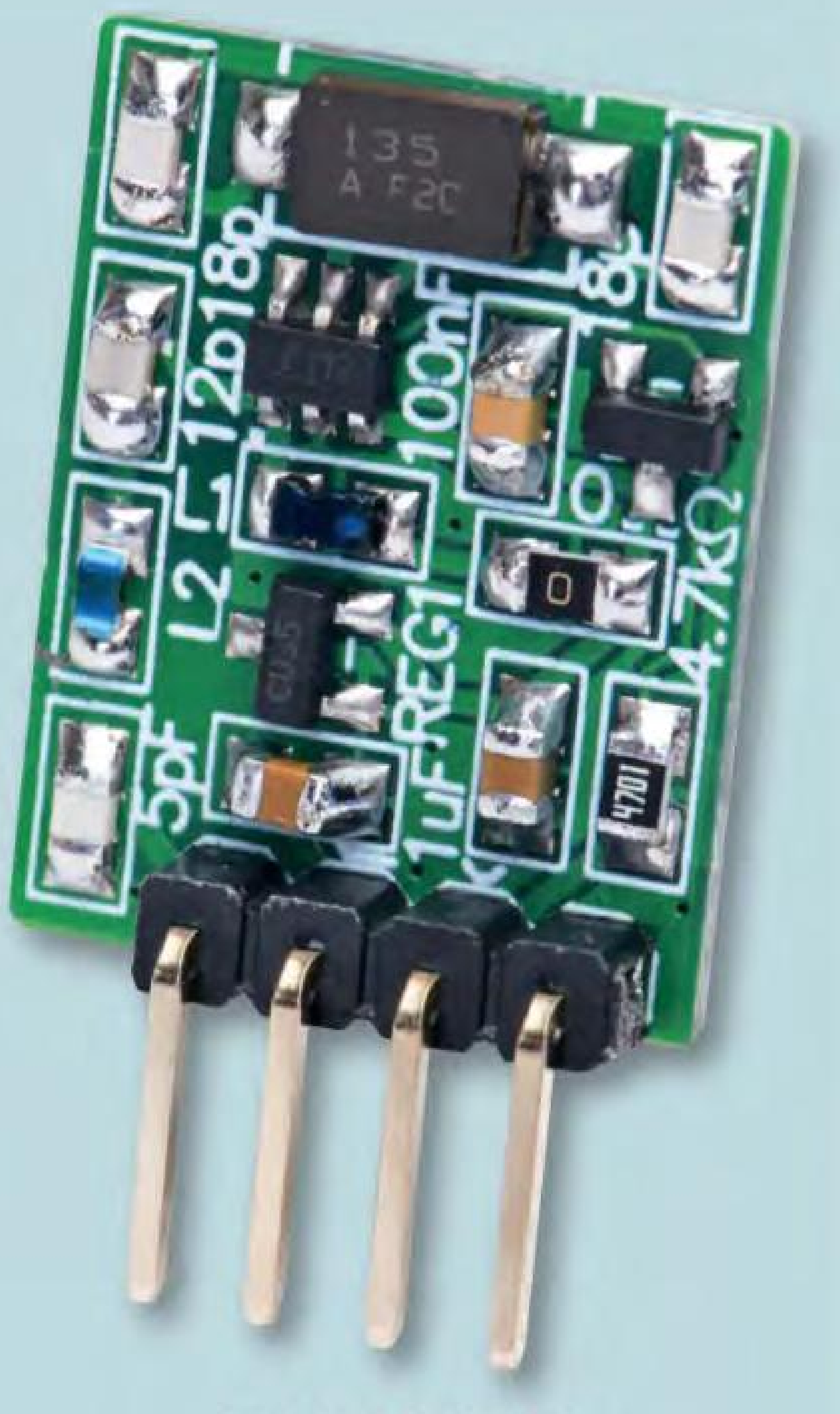
SSB Shortwave Receiver

With digital tuning, USB/LSB support and a squelch control.

SILICON CHIP
Tool Timer v12 Nov 2024
CAUTION 240VAC
P3,4 = in, in -> 10min
P3,4 = in, off -> 20min
P3,4 = off, in -> 30min
P3,4 = off, off -> 60min
PIC16F15
BC338
1N4148

Tool Safety Timer

Automatically switch off a soldering iron or similar if you walk away and forget.



DIY 433MHz Transmitter Module

With better performance than many pre-made modules.

On sale 2nd of April, 2026

Content may be subject to change

Practical Electronics

Subscriptions hotline 07918 614662

To subscribe or renew online, paying by credit/debit card or PayPal, please visit our shop at: www.electronpublishing.com

For those wishing to pay by direct debit, cheque or over the telephone, please call 'Select Publisher Services' on 01202 087631, or send a cheque (payable to: '**Select Publisher Services Ltd**') with your details to:

Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 07918 614662 or visit: www.electronpublishing.com

Published on approximately the first Thursday of each month by Silicon Chip Publications, PO Box 194 Matraville NSW 2036 Australia. Printed in England by The Manson Group, Reynolds House, 8 Porters Wood, Valley Road Industrial Estate, St Albans AL3 6PZ. Distributed by Seymour, 86 Newman St, London W1T 3EX. Subscriptions UK: £33.99 (6mo); £59.99 (12mo.); £114.99 (2yrs). EUROPE: airmail service, £38.99 (6mo); £69.99 (12mo); £129.99 (2yrs). REST OF THE WORLD: airmail service, £46.99 (6mo); £84.99 (12mo); £164.99 (2yrs). Make payments to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the conditions that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter.

Did you know our online shop now sells the current issue of *PE* for £5.99 inc. p&p?

Practical Electronics

The UK's premier electronics, computing and maker magazine

Max's Cool Beans Digital Etch A Sketch & the classic Snake game

Teach-In 12 Part 6: RF testing and measurement

Audio Out All-pass filters: unsung heroes

Circuit Surgery Stabilising op amps using output resistors

Power Rail Probe shows supply noise and ripple

Versatile Battery Checker

Test batteries, large and small!

Pico 2 Audio Analyser

PLUS!

Techno Talk: optical MEMS microphones and nanoscale sensors

Our new **Power Electronics** series will cover DC/DC, AC/DC and DC/AC conversion, power factor correction (PFC) and more

Apr 2026 £5.99

www.electronpublishing.com | @practicalelec | practicalelectronics

Practical Electronics

Audio Out Silent stomp switch circuit

Circuit Surgery Op amp capacitive loading and stability

A Frequency Divider configured using USB

Mar 2026 £5.99

@practicalelec | practicalelectronics

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

www.electronpublishing.com

5V Operation in Any Condition

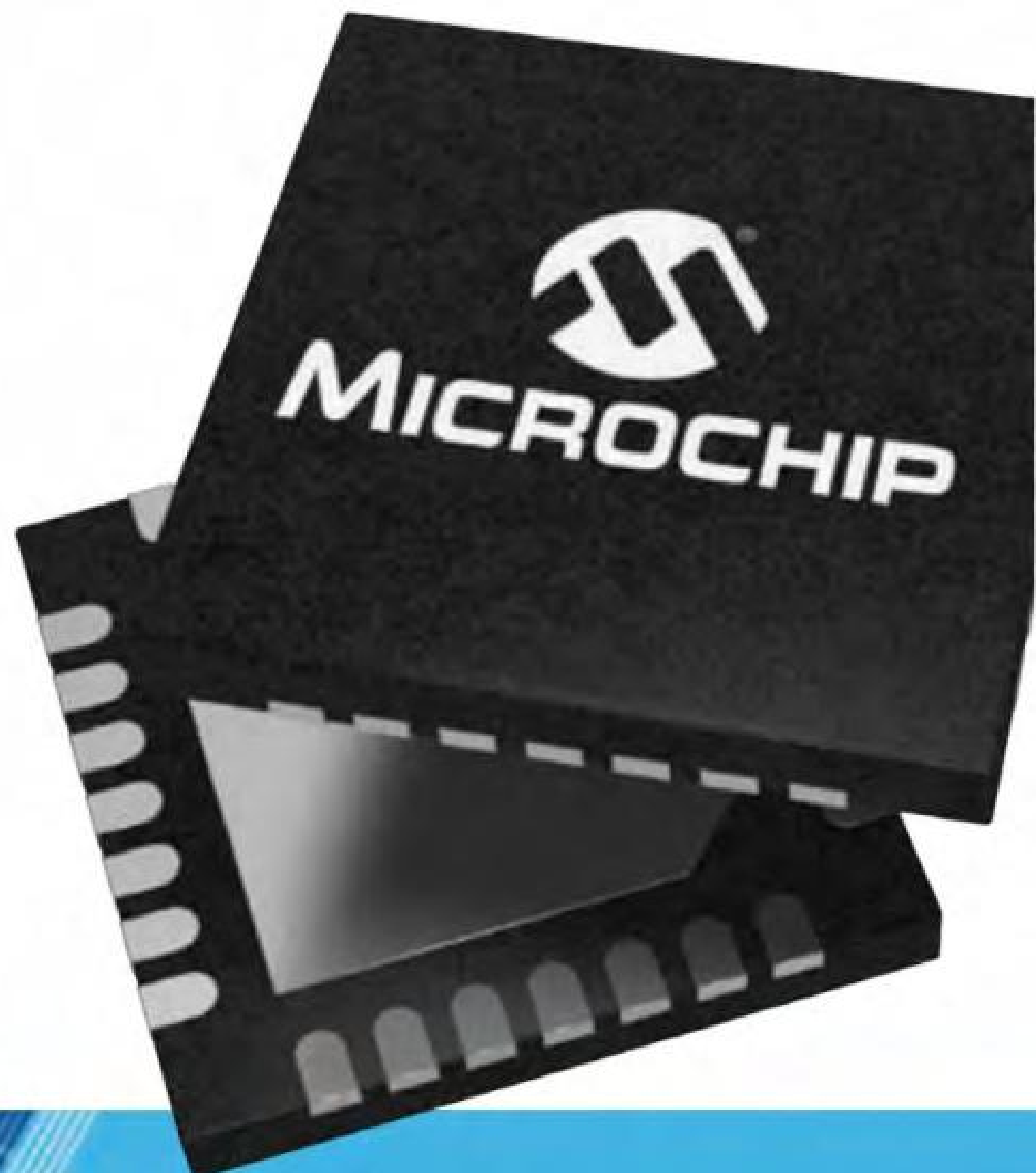
Efficient Performance Without Power Penalties

The PIC32CM PL10 microcontrollers redefines what's possible for engineers seeking an ideal balance between simplicity and performance. Built on the ARM Cortex-M0+ core, these microcontrollers deliver true 5V operation, rare among 32-bit MCUs, ensuring exceptional noise immunity for industrial, appliance and automotive applications. With advanced touch sensing, ultra-low power consumption and seamless support for familiar development tools, the PIC32CM PL10 bridges simplicity and performance with robust capacitive touch capability and reliable 5V support.

Key Features

- True 5V Operation: Robust performance in noisy environments
- Advanced Touch Sensing: Peripheral Touch Controller supports high channel count and resists interference
- Ultra-Low Power Modes: Sleepwalking and low standby current consumption extend battery life
- Easy Migration: Designed for 8-bit users to upgrade with no growing pains
- Familiar Development Tools: Compatible with Microsoft® Visual Studio Code (VS Code®) and MPLAB® Code Configurator, and supported by partner toolchains like IAR Embedded Workbench, Keil and Segger.
- Competitive Price Point: High-end features without premium cost

Upgrade your next design with the PIC32CM PL10 and experience 32-bit performance without the complexity.



MICROCHIP

microchip.com/pic32cm-pl10

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2026 Microchip Technology Inc. All rights reserved.
MEC2639A-UK-01-26